

# NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



## THESIS

### FINITE ELEMENT MODEL UPDATING AND DAMAGE DETECTION USING ARTIFICIAL BOUNDARY CONDITIONS

by

Christopher P. DeGregory

March 1999

Thesis Advisor:

Joshua H. Gordis

Approved for public release; distribution is unlimited.

1 99904 150 16

DTIC QUALITY INSPECTED 4

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Finite Element Model Updating and Damage Detection Using Artificial Boundary Conditions		5. FUNDING NUMBERS	
6. AUTHOR(S) DeGregory, Christopher P.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed here are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT ( <i>maximum 200 words</i> )      An experimental test of a structure results in a database of frequency response functions from which natural frequencies and mode shapes are identified for the given boundary conditions of the test. The natural frequencies of a system under a variety of boundary conditions can be identified by applying artificial boundary conditions (ABCs) at measurement locations and obtaining the frequency response function for the unrestrained degrees of freedom. These frequencies are found without any physical alterations of the test boundary conditions. Use of the ABCs in sensitivity-based model updating and damage detection in conjunction with baseline data provides improved error localization and results in a more accurate finite element model.			
14. SUBJECT TERMS FINITE ELEMENT MODEL		15. NUMBER OF PAGES 94	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18 298-102



**FINITE ELEMENT MODEL UPDATING AND DAMAGE DETECTION  
USING ARTIFICIAL BOUNDARY CONDITIONS**

Christopher P. DeGregory  
Lieutenant, United States Navy  
B.S. Civil Engineering, Worcester Polytechnic Institute, 1991


Submitted in partial fulfillment of the  
Requirements for the degree of

**MASTER OF SCIENCE IN MECHANICAL ENGINEERING**

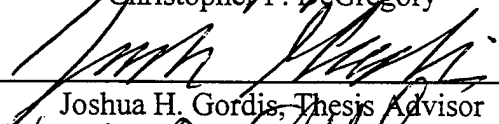
from the


**NAVAL POSTGRADUATE SCHOOL  
March 1999**

Author:

  
\_\_\_\_\_  
Christopher P. DeGregory

Approved by:

  
\_\_\_\_\_  
Joshua H. Gordis, Thesis Advisor

  
\_\_\_\_\_  
Terry R. McNelley, Chairman  
Department of Mechanical Engineering



## **ABSTRACT**

An experimental test of a structure results in a database of frequency response functions from which natural frequencies and mode shapes are identified for the given boundary conditions of the test. The natural frequencies of a system under a variety of boundary conditions can be identified by applying artificial boundary conditions (ABCs) at measurement locations and obtaining the frequency response function for the unrestrained degrees of freedom. These frequencies are found without any physical alterations of the test boundary conditions. Use of the ABCs in sensitivity-based model updating and damage detection in conjunction with baseline data provides improved error localization and results in a more accurate finite element model.



## TABLE OF CONTENTS

I. INTRODUCTION.....	1
II. THEORY.....	5
A. OMITTED-COORDINATE SET .....	5
B. EXACT DYNAMIC REDUCTION AND FRF MATRICES .....	6
C. DRIVING POINT FREQUENCY RESPONSE FUNCTION .....	7
III. ABC CONFIGURATION FREQUENCIES .....	11
A. DRIVING POINT ANTIRESONANCES DEFINE ABC FREQUENCIES FOR A SINGLE A-SET COORDINATE .....	11
B. ABC CALCULATIONS FOR A 2 DOF SYSTEM.....	12
C. ABC FREQUENCY CALCULATIONS FOR FREE-FREE BEAM	14
IV. ABC CONFIGURATION IN SENSITIVITY-BASED UPDATING.....	19
A. SENSITIVITY MATRIX DEFINED .....	19
B. IMPROVED CONDITIONING USING ABC SENSITIVITIES.....	21
C. COMPUTER SIMULATED DAMAGE DETECTION USING ABC SYSTEM SENSITIVITIES .....	21
1. Damage Detection: Example 1 .....	22
2. Damage Detection: Example 2 .....	23
3. Damage Detection: Example 3 .....	25
4. Damage Detection: Example 4 .....	25
5. Damage Detection: Example 5 .....	28
6. Damage Detection: Example 6 .....	29
V. EXPERIMENTAL APPLICATION.....	31



A. EQUIPMENT SETUP .....	31
B. DATA COLLECTION.....	33
C. O-SET FREQUENCIES USING EXPERIMENTAL DATA .....	33
D. DAMAGE DETECTION USING EXPERIMENTAL DATA.....	43
VI. CONCLUSIONS AND RECOMMENDATIONS .....	47
A. CONCLUSIONS.....	47
B. RECOMMENDATIONS .....	48
APPENDIX MATLAB CODE .....	49
A. MAIN PROGRAMS .....	49
B. SUPPORT PROGRAMS .....	64
C. FUNCTIONS CALLED BY MAIN & SUPPORT PROGRAMS.....	73
LIST OF REFERENCES .....	77
INITIAL DISTRIBUTION LIST .....	79

## LIST OF FIGURES

Figure 3.1 2 DOF System .....	11
Figure 3.2 2 DOF System with DOF #1 restrained to ground .....	12
Figure 3.3 $H_{11}(\Omega)$ versus $\Omega$ .....	13
Figure 3.4 $H_{11}^{-1}(\Omega)$ versus $\Omega$ .....	14
Figure 3.5 Free-free beam.....	15
Figure 3.6 Driving Point FRF $H_{11}(\Omega)$ for Free-Free beam.....	15
Figure 3.7 $H_{11}^{-1}(\Omega)$ for free-free beam (a-set DOFs: 1,5,9,13,17,21).....	16
Figure 3.8 ABC Configuration .....	17
Figure 3.9 $H_{11}^{-1}(\Omega)$ for free-free beam (a-set DOFs: 1, 17).....	18
Figure 3.10 ABC Configuration - a-set: [1 17].....	18
Figure 4.1 Damage Detection - Example 1.....	23
Figure 4.2 Damage Detection - Example 2.....	24
Figure 4.3 Damage Detection - Example 3.....	26
Figure 4.4 Damage Detection - Example 4.....	27
Figure 4.5 Damage Detection - Example 5.....	28
Figure 4.6 Damage Detection - Example 6.....	29
Figure 5.1 Laboratory Setup .....	32
Figure 5.2 $H_{11}(\Omega)$ for free-free beam.....	34
Figure 5.3 ABC Configuration: Node 1 restrained to ground .....	35
Figure 5.4 $H_{aa}^{-1}(\Omega)$ for Node 1 restrained to ground .....	36
Figure 5.5 ABC Configuration: Nodes 1 & 2 restrained to ground.....	37
Figure 5.6 $H_{aa}^{-1}(\Omega)$ for Nodes 1,2 restrained to ground.....	37
Figure 5.7 ABC Configuration: Nodes 1,4 restrained to ground.....	38
Figure 5.8 $H_{aa}^{-1}(\Omega)$ for Nodes 1,4 restrained to ground.....	38
Figure 5.9 ABC Configuration: Nodes 1,2,3 restrained to ground.....	39
Figure 5.10 $H_{aa}^{-1}(\Omega)$ for Nodes 1,2,3 restrained to ground.....	39
Figure 5.11 ABC Configuration: Nodes 1,2,3,4 restrained to ground.....	40

Figure 5.12 $H_{aa}^{-1}(\Omega)$ for Nodes 1,2,3,4 restrained to ground.....	40
Figure 5.13 ABC Configuration: Nodes 1,2,3,4,5 restrained to ground.....	41
Figure 5.14 $H_{aa}^{-1}(\Omega)$ for Nodes 1,2,3,4,5 restrained to ground.....	41
Figure 5.15 ABC Configuration: Nodes 1,3,5 restrained to ground.....	42
Figure 5.16 $H_{aa}^{-1}(\Omega)$ for Nodes 1,3,5 restrained to ground.....	42
Figure 5.17 Error localization using FE frequencies .....	43
Figure 5.18 Error localization using test frequencies .....	44

## LIST OF TABLES

Table 4.1	Example 1: FE and Test Frequencies and Error .....	22
Table 4.2	Example 2: FE and Test Frequencies and Error .....	24
Table 4.3	Example 3: FE and Test Frequencies and Error .....	25
Table 4.4	Example 4: FE and Test Frequencies and Error .....	27
Table 5.1	FE & Test Frequencies for free-free beam .....	34
Table 5.2	FE & Test Frequencies for Node 1 restrained to ground .....	35
Table 5.3	FE & Test Frequencies for Nodes 1,2 restrained to ground .....	37
Table 5.4	FE & Test Frequencies for Nodes 1,4 restrained to ground .....	38
Table 5.5	FE & Test Frequencies for Nodes 1,2,3 restrained to ground .....	39
Table 5.6	FE & Test Frequencies for Nodes 1,2,3,4 restrained to ground .....	40
Table 5.7	FE & Test Frequencies for Nodes 1,2,3,4,5 restrained to ground .....	41
Table 5.8	FE & Test Frequencies for Nodes 1,3,5 restrained to ground .....	42



## ACKNOWLEDGMENTS

I would like to express my sincerest appreciation to Professor Joshua H. Gordis for his dedicated support and undying patience throughout this endeavor.

I would also like to thank Frank Franzen for his craftsmanship and Tom Christian for never giving up on the installation of *I-DEAS* and making the laboratory experiment a success.

## I. INTRODUCTION

In general, a finite element (FE) model is defined by a large number of physical parameters, but a modal test of a structure results in a small number of modal parameters which are used to modify the model parameters. It is always crucial, then, to improve or "update" a finite element model in order to produce a reliable, confident prediction of the structural response of a system. Adjusting the parameters that define the model can reduce inaccuracies in a FE model. These parameters include, but are not limited to dimensional properties, moduli of elasticity, and density. If the parameters in the model can be adjusted for closer agreement with the measured parameters (identified in a modal test), then the model will provide a more accurate representation of dynamic response. It is important to remember that no FE model can exactly predict the dynamic response of a real structure, but can still provide reasonable and reliable results. This is the goal of model updating.

One problem that has been historically associated with finite element model verification has been in measuring enough information. Procedures have been developed for measuring larger experimental databases. One such method that has been developed is known as "Perturbed Boundary Condition" testing. [Ref. 4] In this procedure, selected boundary conditions of a system under test are drastically changed and the system is retested. This method is effective, however the procedure requires that physical modifications be made to the structure and additional tests are necessary for each modification made. The result is increased time, increased work and most importantly in today's world, increased cost.

A more efficient method is to identify additional and distinct mode frequencies from the same modal test that was performed to identify the mode frequencies of the standard system, without the need for physical modification of the structure. The additional frequencies and mode shapes correspond exactly to the mode frequencies found when combinations of measured coordinates are constrained to ground. These additional frequencies are therefore associated with different boundary conditions for the structure and no physical change in the boundary conditions has been made. [Ref. 1]

Because the boundary conditions are not actually applied, they are termed "artificial boundary conditions", or "ABCs." For a single degree of freedom system the driving point antiresonance frequencies of a frequency response function (FRF) correspond to the frequencies of the structure with the driving point degree-of-freedom restrained to ground. The ABCs are imposed on the FE model and correspond to ideal constraints. With only one experimental database, this scheme yields a separate FE model for each ABC configuration. With the exception of the imposed boundary conditions, each model is identical and can be used to generate sensitivity data. In a set of spatially incomplete frequency response function data, the ABCs are the boundary conditions that define an "omitted coordinate system (OCS)," or "o-set." In performing a vibration test, a choice is made as to the set of coordinates to instrument with response transducers. This is the "analysis set," or the "a-set." With the choice of an a-set, the o-set is defined as the complimentary set of coordinates. For an FE model, the o-set is of finite dimension, but with respect to the test system, this set is of infinite dimension. The natural mode frequencies of the o-set systems appear as peaks in the impedance spectra. [Ref. 2]



ABC configuration frequencies are available from any set of test data due to the fact that a spatially incomplete FRF matrix is identically equal to the FRF matrix that is calculated from the exact dynamic reduction. [Ref. 1]

In addition to providing a larger number of frequencies for a system using one experimental database, the ABC also provides a means to eliminate or greatly reduce ill-conditioning in sensitivity equations. With the system artificially restrained at various measured coordinates, the columns of the sensitivity matrix which are found from the ABC configuration are linearly independent of the columns calculated from the baseline configuration. This reduces the difficulties in determining the parameters that are in error. Most importantly, in order to more accurately update the FE model, this configuration improves error localization.



## II. THEORY

### A. OMITTED-COORDINATE SET

The equation for the steady-state harmonic response for a full order model (without damping) is

$$\begin{bmatrix} \mathbf{k}_{aa} & \mathbf{k}_{ao} \\ \mathbf{k}_{oa} & \mathbf{k}_{oo} \end{bmatrix} - \Omega^2 \begin{bmatrix} \mathbf{m}_{aa} & \mathbf{m}_{ao} \\ \mathbf{m}_{oa} & \mathbf{m}_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_a \\ \mathbf{x}_o \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_a \\ \mathbf{f}_o \end{Bmatrix} \quad (2.1)$$

where  $\Omega$  is the frequency of harmonic excitation and  $\mathbf{f}$  is a vector of excitations consistent with  $\mathbf{x}$  and  $\mathbf{k}$  and  $\mathbf{m}$  are the stiffness and mass matrices, respectively. The subscript "a" refers to the "a-set" and the subscript "o" refers to the "o-set". Equation 2.1 is also known as the impedance model and can be redefined as:

$$\begin{bmatrix} \mathbf{Z}_{aa} & \mathbf{Z}_{ao} \\ \mathbf{Z}_{oa} & \mathbf{Z}_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_a \\ \mathbf{x}_o \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_a \\ \mathbf{f}_o \end{Bmatrix} \quad (2.2)$$

where  $\mathbf{Z}(\Omega)$  is the system impedance matrix evaluated at the forcing frequency,  $\Omega$ .

Assuming that  $\{\mathbf{f}_o\} = \{0\}$  (static constraint - no excitation acting on the omitted coordinates), the exact relationship between the o-set and a-set coordinates is as follows:

$$\{\mathbf{x}_o\} = [\mathbf{I} - \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oo}]^{-1} [-\mathbf{k}_{oo}^{-1} \mathbf{k}_{oa} + \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oa}] \{\mathbf{x}_a\} \quad (2.3a)$$

or

$$\{\mathbf{x}_o\} = [-\mathbf{Z}_{oo}]^{-1} [\mathbf{Z}_{oa}] \{\mathbf{x}_a\} \quad (2.3b)$$

The origin of the o-set system results from Eq. (2.3a). By definition, the bracketed inverse term is

$$\left[ \mathbf{I} - \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oo} \right]^{-1} = \frac{1}{\text{Det} \left[ \mathbf{I} - \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oo} \right]} \text{Adj} \left[ \mathbf{I} - \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oo} \right] \quad (2.4)$$

where  $\text{Det}[\bullet]$  indicates the determinant and  $\text{Adj}[\bullet]$  indicates the adjoint matrix. From Eq. (2.4), it is clear that the bracketed inverse term does not exist at those frequencies  $\Omega$  which satisfy

$$\text{Det} \left[ \mathbf{I} - \Omega^2 \mathbf{k}_{oo}^{-1} \mathbf{m}_{oo} \right] = 0 \quad (2.5)$$

Therefore the relationship between the a-set and o-set does not exist at those frequencies. The frequencies that satisfy Eq. (2.5) are the eigenvalues of the system defined by  $[\mathbf{k}_{oo}]$  and  $[\mathbf{m}_{oo}]$ , the o-set system. This system is obtained by fully constraining all coordinates in the a-set to ground. The use of spatially incomplete FRF data in identification of the o-set mode frequencies implies the use of dynamic reduction applied to the structure under test. [Ref. 2]

## B. EXACT DYNAMIC REDUCTION AND FRF MATRICES

By equipping a structure with a finite number of response transducers, a reduced order model is defined, where the impedance of the reduced order model is not linearly dependent on the impedance of the full order model. The "full-order", exact FRF model of a structure is a FRF matrix of infinite dimension,

$$\mathbf{H}^x = \begin{bmatrix} \mathbf{H}_{aa} & \mathbf{H}_{ao} \\ \mathbf{H}_{oa} & \mathbf{H}_{oo} \end{bmatrix} \quad (2.6)$$

where the number of coordinates in the o-set is infinite. The FRF matrix measured in a test is a matrix partition extracted from the infinite dimension matrix,

$$\overline{\mathbf{H}}^x = \mathbf{H}_{aa} \quad (2.7)$$

where the overbar notation indicates a reduced model. The matrix partition defined by Eq. (2.7) represents a structural dynamic model that has been reduced using exact dynamic reduction. From the partitioned inverse relation of the FRF matrix to its associated impedance,  $\mathbf{ZH} = \mathbf{I}$ . [Ref. 2]

$$\mathbf{H}_{aa} = (\mathbf{Z}_{aa} - \mathbf{Z}_{ao}\mathbf{Z}_{oo}^{-1}\mathbf{Z}_{oa})^{-1} \quad (2.8)$$

The presence of the o-set system dynamics in the term  $\mathbf{Z}_{oo}^{-1}$  is seen in Eq. (2.8). The formula for the matrix inverse seen in Eq. (2.4) is applicable. Since every element in  $\mathbf{Z}_{oo}^{-1}$  is singular at the natural frequencies of the o-set, Eq. (2.8) shows that elements of  $\mathbf{H}_{aa}^{-1}$  will be singular at the o-set natural frequencies [Ref. 1]. Since the measured FRF matrix implicitly defines a dynamically reduced impedance model, an order  $n$  FRF matrix is calculated from the FE model, and the partition of this matrix which corresponds to the coordinates measured in a test is extracted (the a-set). The reduced model retains the entire modal content of the original model.

### C. DRIVING POINT FREQUENCY RESPONSE FUNCTION

Starting again with the equation of steady state forced response for a linear structural dynamic system, Eq. (2.2), rewritten (with damping) as

$$[\mathbf{k} - \Omega^2\mathbf{m} - j\Omega\mathbf{C}]\{\mathbf{x}\} = \{\mathbf{f}\} \quad (2.9)$$

where  $[\mathbf{k} - \Omega^2\mathbf{m} - j\Omega\mathbf{C}] = [\mathbf{Z}]$  and  $\mathbf{C}$  is the damping matrix. Transferring from physical to modal coordinates let

$$\{\mathbf{x}\} = [\Phi]\{\mathbf{q}\} \quad (2.10)$$

Rewriting Eq. (2.9),

$$[\mathbf{Z}][\Phi]\{\mathbf{q}\} = \{\mathbf{f}\} \quad (2.11)$$

Premultiply by  $[\Phi]^T$ ,

$$[\Phi]^T [\mathbf{Z}][\Phi]\{\mathbf{q}\} = [\Phi]^T \{\mathbf{f}\} \quad (2.12a)$$

$$[\Phi^T \mathbf{k} \Phi - \Omega^2 \Phi^T \mathbf{m} \Phi + j\Omega \Phi^T \mathbf{C} \Phi]\{\mathbf{q}\} = \{\mathfrak{F}\} \quad (2.12b)$$

Using orthogonality and assuming proportional damping,

$$[\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i]\{\mathbf{q}\} = \{\mathfrak{F}\} \quad (2.13)$$

where  $\omega_i$  is the natural frequency of the  $i^{\text{th}}$  mode,  $\zeta$  is the damping ratio and

$[\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i]$  is a diagonal matrix. This matrix is inverted to find the modal

frequency response:

$$\{\mathbf{q}\} = \left[ \frac{1}{\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i} \right] \{\mathfrak{F}\} \quad (2.14)$$

Transforming back into physical coordinates by premultiplying by  $[\Phi]$  and using

$\{\mathfrak{F}\} = [\Phi]^T \{\mathbf{f}\}$ , the modal decomposition of the FRF matrix in physical coordinates is as

follows:

$$\{\mathbf{x}\} = [\Phi] \left[ \frac{1}{\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i} \right] [\Phi]^T \{\mathbf{f}\} \quad (2.15)$$

where  $[\mathbf{H}(\Omega)] = [\Phi] \left[ \frac{1}{\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i} \right] [\Phi]^T$ .  $[\mathbf{H}(\Omega)]$  can be written:

$$[\mathbf{H}(\Omega)] = \sum_{p=1}^{\text{NDOF}} \frac{\{\phi^p\} \{\phi^p\}^T}{\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i} \quad (2.16a)$$

or any element,

$$\mathbf{H}_{ij}(\Omega) = \sum_{p=1}^{\text{NDOF}} \frac{\{\phi_i^p\}\{\phi_j^p\}}{\omega_i^2 - \Omega^2 + 2j\zeta\Omega\omega_i} \quad (2.16b)$$

This driving point FRF will be used in subsequent chapters in defining the ABC natural frequencies.

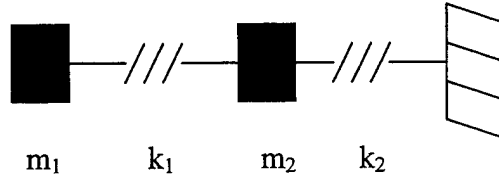




### III. ABC CONFIGURATION FREQUENCIES

#### A. DRIVING POINT ANTIRESONANCES DEFINE ABC FREQUENCIES FOR A SINGLE A-SET COORDINATE

The following example using a basic 2 degree-of-freedom (DOF) system, shows that driving point antiresonances correspond to the natural frequencies of the structure with the driving point DOF constrained to ground.



**Figure 3.1** 2 DOF system

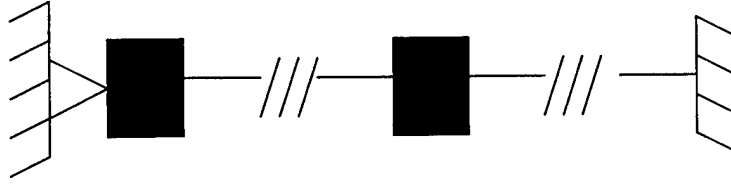
From Eq. (2.16b), the undamped driving point FRF is given by

$$\mathbf{H}_{ij}(\Omega) = \sum_{r=1}^{NDOF} \frac{\{\phi_i^r\} \{\phi_j^r\}}{\omega_r^2 - \Omega^2} \quad (3.1)$$

where  $\phi_i$  is a mass normalized mode shape element,  $\omega_r$  is the  $r^{\text{th}}$  natural frequency, and  $\Omega$  is the forcing frequency. The frequency of the anti-resonance of  $H_{11}(\Omega)$  is given by

$$\Omega_{\text{anti-res}}^2 = \frac{R_{11}^1 \omega_2^2 + R_{11}^2 \omega_1^2}{R_{11}^1 + R_{11}^2} \quad (3.2)$$

where the modal residue is given by  $R_{ij}^r = \phi_i^r \phi_j^r$ . It can be demonstrated that this frequency is identical to the natural frequency of the system in Figure 3.1, with the driving point DOF constrained to ground as in Figure 3.2.

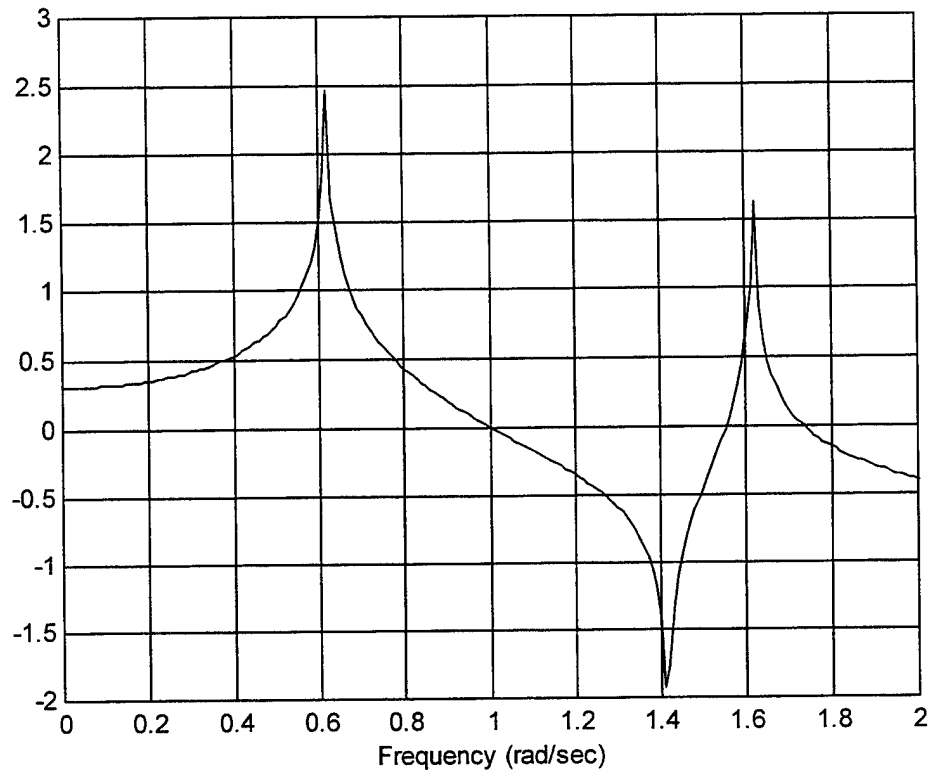


**Figure 3.2** 2 DOF system with DOF #1 restrained to ground.

Using a simplified numerical example, if the values for Figure 3.1 are set as  $m_1=m_2=1.0$  and  $k_1=k_2=1.0$ , the frequency of the single antiresonance is  $\Omega_{\text{anti-res}} = \sqrt{2}$  rad/sec, which is identically equal to the single natural frequency of the ABC system seen in Figure 3.2, which is  $\omega = \sqrt{2}$  rad/sec.

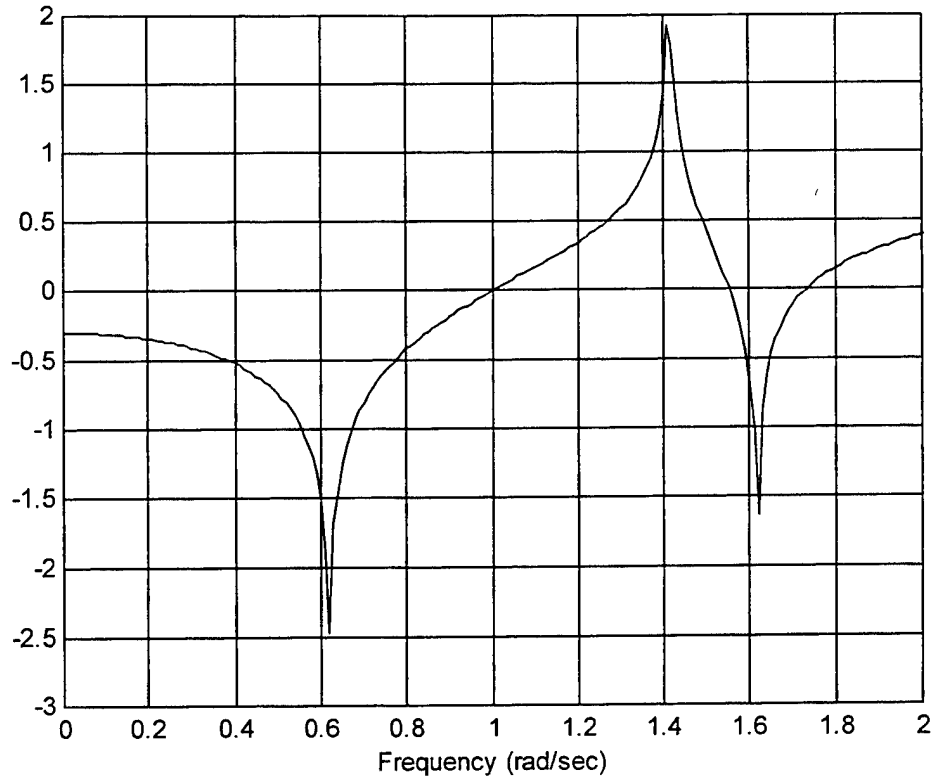
#### **B. ABC CALCULATIONS FOR A 2 DOF SYSTEM**

The ABC frequencies are calculated using Eq. (2.8). A spatially incomplete FRF matrix is generated and inverted at each frequency. In a plot of the elements of the resulting impedance matrix versus frequency, the singular frequencies are also the ABC system frequencies. To show an example, the 2 DOF system shown in Figure 3.1 is used to calculate the driving point FRF  $H_{11}(\Omega)$ . The system contains two modes and the frequencies of these modes are  $\omega_1 = 0.618$  rad/sec ( $f_1 = 0.0984$  Hz) and  $\omega_2 = 1.618$  rad/sec ( $f_2 = 0.2575$  Hz), respectively. A plot of  $H_{11}(\Omega)$  is shown in Figure 3.3 and these modes can be seen on the plot. In addition to these two modes, it can also be seen that the anti-resonance frequency, as computed in Eq. (3.2), is  $\Omega_{\text{anti-res}} = \sqrt{2} = 1.414$  rad/sec.



**Figure 3.3**  $H_{11}(\Omega)$  versus  $\Omega$

Using Eq. (2.8), the ABC system frequency can be found by calculating  $H_{aa}^{-1}(\Omega)$  where the a-set is defined as DOF #1, and  $H_{aa}^{-1}(\Omega)$  is the scalar inverse of  $H_{11}(\Omega)$ .  $H_{aa}^{-1}(\Omega)$  is plotted versus  $\Omega$  in Figure 3.4. The single singular frequency corresponds to the anti-resonance frequency for this case ( $\Omega_{\text{anti-res}} = 1.414$  rad/sec). This singular frequency is also the natural frequency of the ABC system that was obtained by constraining DOF #1 to ground (as in Figure 3.2). Most importantly, this simple example demonstrates the power and usefulness of Eq. (2.8) and shows that driving point antiresonances correspond



**Figure 3.4**  $H_{11}^{-1}(\Omega)$  versus  $\Omega$   
(Natural Frequency of ABC system)


directly to the natural frequencies of the system with the driving point coordinate constrained to ground. [Ref. 1]

### C. ABC FREQUENCY CALCULATIONS FOR A FREE-FREE BEAM

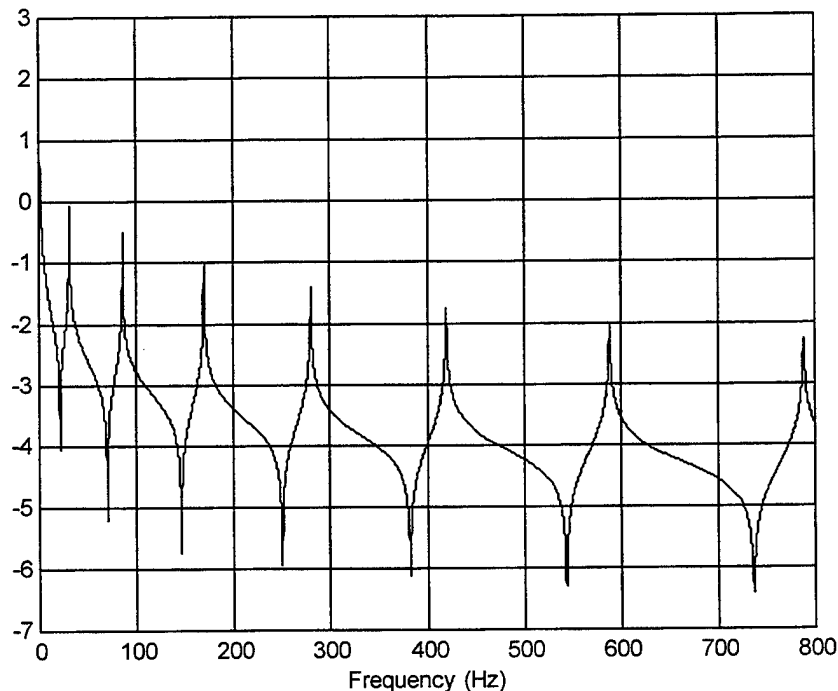
To further demonstrate the usefulness of the ABC technique, a free-free beam is examined. The model consists of 10 beam elements and each element contains two nodes. The nodes have 2 degrees-of-freedom, translational and rotational, giving the system a total of 22 DOF. The odd-numbered DOF are the translational degrees and the even-numbered DOF are the rotational degrees. The properties of the beam are as follows: length = 60 inches,  $EI = 5.5 \times 10^5$  lbf-in<sup>2</sup>,  $\rho = 0.283$  lbf/in<sup>2</sup>, and cross-sectional

area = 0.75 in<sup>2</sup>. Response transducers (accelerometers) are placed at every other node, at the translational DOFs #1,5,9,13,17, and 21. This model is shown in Figure 3.5



**Figure 3.5** Free-free beam - Transducer: 

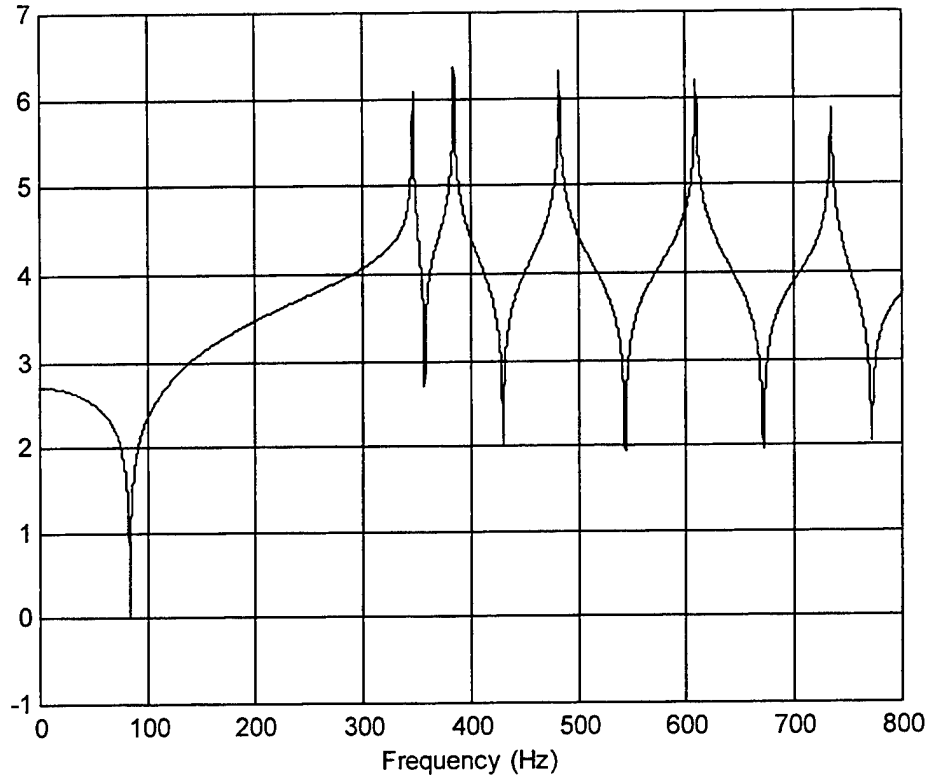
The set of coordinates chosen to be instrumented with response transducers is the a-set, in this example defined as [1 5 9 13 17 21]. It can be assumed by the given setup, that excitation is applied at each of the a-set DOFs, resulting in a 6 by 6 matrix. The impedance matrix,  $H_{aa}^{-1}(\Omega)$ , the scalar inverse of  $H_{11}(\Omega)$ , is calculated at excitation frequencies from 0-800 Hz. Figure 3.6 shows the driving point FRF for the system.



**Figure 3.6** Driving Point FRF  $H_{11}(\Omega)$  for Free-Free Beam  
(a-set DOFs: 1, 5, 9, 13, 17, 21)

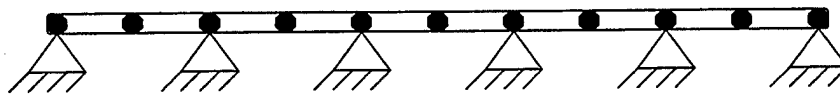
The ABC system frequencies are found by calculating the impedance matrix,  $H_{aa}^{-1}(\Omega)$ .

By plotting the 1,1 element ( $H_{11}^{-1}(\Omega)$ ) of the matrix, the ABC system frequencies can be seen (Figure 3.7).



**Figure 3.7**  $H_{11}^{-1}(\Omega)$  for Free-Free Beam  
(a-set DOFs: 1, 5, 9, 13, 17, 21)

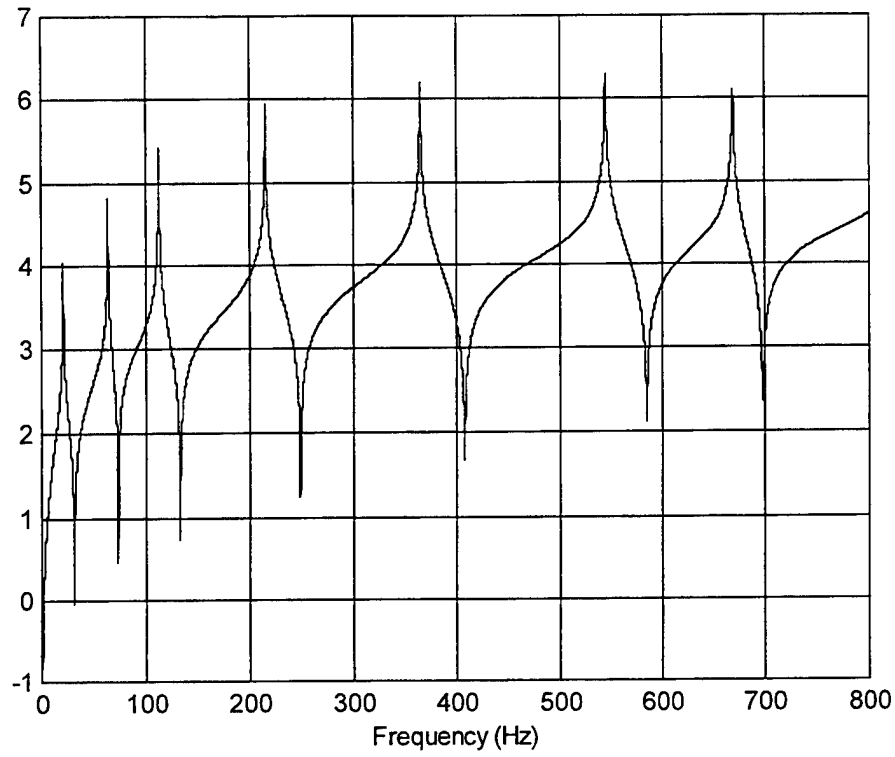
These ABC frequencies correspond exactly to the natural frequencies of the system with all of the measured coordinates (a-set) constrained to ground. This configuration is shown in Figure 3.8. Confirming the results, the natural frequencies of a beam with pin restraints in the locations shown in Figure 3.8 are calculated to be: 346.70, 385.13,



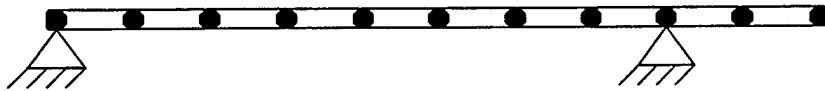
**Figure 3.8** ABC Configuration – a-set: [1 5 9 13 17 21]  
(Measured coordinates restrained to ground)

482.73, 610.23, and 735.11 Hz. These frequencies correspond exactly to the peaks shown in Figure 3.7.

A common vibrational test involves two shakers placed to excite all modes in the bandwidth of interest. To simulate this, the next ABC configuration set will be DOFs 1 and 17, defined as the translations at nodes 1 and 9. Performing the same calculations as above, the impedance matrix,  $H_{aa}^{-1}(\Omega)$ , is obtained. Looking at the 1,1 element of this matrix, an additional set of ABC frequencies is found. A plot of  $H_{11}^{-1}(\Omega)$  over the range 0-800 Hz is shown in Figure 3.9. The equivalent system is a beam with pin restraints at node 1 and node 9. The natural frequencies of this system are calculated to be 20.45, 63.57, 112.30, 215.05, 364.61, 544.16, and 669.70 Hz. These frequencies correspond exactly to the peaks shown in Figure 3.9. The equivalent system is shown in Figure 3.10.



**Figure 3.9**  $H_{11}^{-1}(\Omega)$  for Free-Free Beam  
(a-set DOFs: 1, 17)



**Figure 3.10** ABC Configuration - a-set: [1 17]  
(Measured coordinates restrained to ground)



## IV. ABC CONFIGURATION IN SENSITIVITY-BASED UPDATING

### A. SENSITIVITY MATRIX DEFINED

As previously mentioned, the disparity in the number of known parameters (measured modal parameters) versus the number of parameters to be adjusted in order to update a FE model defines an underdetermined problem. Another significant difficulty is that the parameters that are in error are unknown. Sensitivity-based updating is used for error localization in order to find those parameters that require adjustment. The ABC configuration frequencies can be used in addition to, or instead of, the standard baseline system mode frequencies in model updating and damage detection. The ABC frequencies correspond to the same structural system as the baseline frequencies, but with different boundary conditions. The governing equation for sensitivity-based updating is

$$\{\Delta\omega\} = [T]\{\Delta DV\} \quad (4.1)$$

where  $\{\Delta\omega\}$  is a vector of natural frequency errors. The errors are the difference between the experimental natural frequencies and the analytical natural frequencies  $\{\omega^x - \omega^a\}$ .

The  $\{\Delta DV\}$  term is the vector of changes to be calculated for specified model parameters, known as "design variables," and  $[T]$  is the sensitivity matrix. This matrix is solved using the eigensensitivity method shown in Ref. [6].  $[T]$  is defined as:

$\{\phi_{\text{baseline}}\}^T [\Delta k] \{\phi_{\text{baseline}}\}$ , where  $\Delta k = [k^x - k^a]$  and  $[T]$  is of size  $m \times p$  where  $m$  is the number of modes and  $p$  is the number of design variables used. Each ABC system defines additional rows of Eq. (4.1), defining the equation,

$$\{\Delta\omega^i\} = [T^i]\{\Delta DV\} \quad (4.2)$$

where  $T_{ij}^k = \partial\omega_i^k / \partial DV_j$ , and  $\omega_i^k$  is the  $i^{\text{th}}$  natural frequency of the  $k^{\text{th}}$  ABC configuration system. Baseline system quantities are identified with the superscript "0" and ABC systems are identified with superscripts from 1 to "k" where k is the total number of ABC systems used. Combining the equations,

$$\begin{Bmatrix} \{\Delta\omega^0\} \\ \{\Delta\omega^1\} \\ \vdots \\ \{\Delta\omega^k\} \end{Bmatrix} = \begin{Bmatrix} [T^0] \\ [T^1] \\ \vdots \\ [T^k] \end{Bmatrix} \{\Delta DV\} \quad (4.3)$$

The degree of coupling between the ABC systems and the baseline system can be adjusted by deleting or retaining individual columns of  $[T^k]$ . Partial coupling can be established by partitioning such that some of the DV's are associated with the baseline system only, some with both the baseline and the ABC system, and some with the ABC system only. [Ref. 1]

Using the ABC system sensitivities helps to eliminate or greatly reduce the problem of poorly conditioned or rank deficient  $[T]$  matrices. Columns of  $[T^0]$  can be replaced with columns of  $[T^k]$  in order to improve conditioning. Two closely spaced elements in a model will always have columns of  $[T^0]$  which are nearly dependent. This prevents the localization of the error and discrimination of the damage between the elements. By replacing a column of  $[T^0]$  with a column of  $[T^k]$ , the columns are no longer dependent and error localization and damage detection is improved, resulting in a more accurately updated FE model.

## B. IMPROVED CONDITIONING USING ABC SENSITIVITIES

Applying the case of a free-free beam, the sensitivity matrix  $[T]$  is generated for the undamaged structure model in order to solve for the changes in the element EI values representing damage, i.e.  $\Delta EI = \Delta DV$ , and

$$\{\omega^0\} = [T^0]\{\Delta EI\} \quad (4.4)$$

To demonstrate, sensitivities will be calculated for all 10 elements and for the first 10 elastic modes resulting in a  $[T^0]$  matrix of size 10 by 10. A calculation of the rank of this matrix reveals that it is, in fact, rank deficient:  $\text{Rank}(T^0) = 5$ . This will not provide a fully determined solution for  $\{\Delta EI\}$ . With excitation applied at DOFs 1 and 17, it is possible to identify the ABC system frequencies with these DOF constrained to ground. (See Figures 3.9 and 3.10) The corresponding  $[T^1]$  matrix is generated and is also of size 10 by 10,

$$\{\omega^1\} = [T^1]\{\Delta EI\} \quad (4.5)$$

with the calculated rank:  $\text{Rank}(T^1) = 10$ .

The  $[T^1]$  matrix is full rank due to the asymmetric ABC configuration.

## C. COMPUTER SIMULATED DAMAGE DETECTION USING ABC SYSTEM SENSITIVITIES

Four examples have been constructed to demonstrate the usefulness of a single ABC system set of frequencies and sensitivity matrix in lieu of the standard system set. The ABC system described by Figures 3.9 and 3.10, a-set: [1 17], is again used. In these four examples, changes in elemental EI's have been user inputted to simulate a damaged structure. Two subsequent examples will show the usefulness of this technique when damage location and magnitude is unknown.

### 1. Damage Detection: Example 1

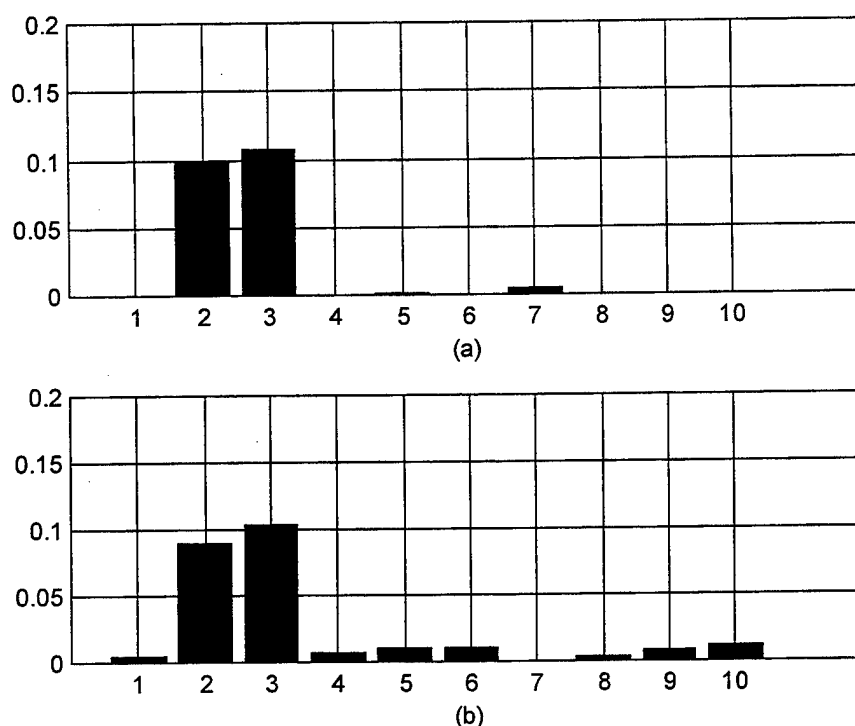
For this example, a 10% reduction in EI is made at elements #2 and #3. The natural frequencies of the undamaged (FE/Analytic) and simulated damaged (Test/Experimental) models are shown in Table 4.1. Using the baseline system, the largest full rank submatrix of  $[T^0]$  is 5 by 10, and the best solution for the damage was the solution using modes 1 to 4. This solution is shown in Figure 4.1(a). The height of the

Mode Number	FE/Analytic (Hz)	Test (Hz)	% Error
1	31.30	31.13	0.5
2	86.30	84.99	1.5
3	169.29	166.37	1.7
4	280.26	276.61	1.3
5	419.78	414.52	1.3
6	588.73	581.00	1.3
7	788.17	779.81	1.1
8	1018.0	1007.8	1.0
9	1264.6	1252.0	1.0
10	1679.4	1662.9	0.9

**Table 4.1** Example 1: FE and Test Frequencies and Error

bars in the plot indicates the magnitude of the error for each element. In this case, we know that the exact solution is 10% in both element #2 and element #3. If the ABC system sensitivities  $[T^1]$  and mode frequencies are used, the largest full rank submatrix is 10 by 10, and the best solution for the damage was the solution using all 10 modes. This solution is shown in Figure 4.1 (b). This was chosen as the first example because in this case, the baseline system provided a reliable result. It should be noted that the ABC system also provided reasonable results (other elements were affected by 1% or less). It

will be seen in the following examples that the baseline system rarely provides reliable results and in many circumstances actually provides results in misleading information.



**Figure 4.1 Damage Detection – Example 1**  
(a) Baseline (b) ABC system

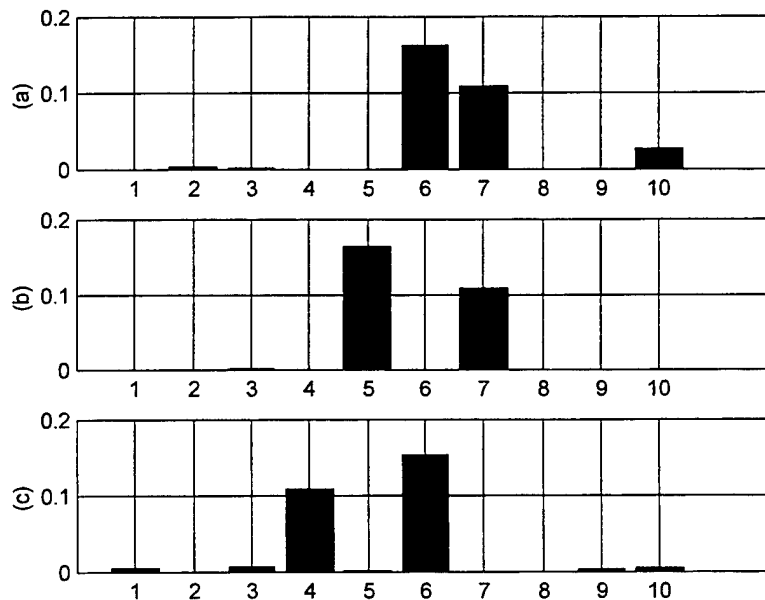
## 2. Damage Detection: Example 2

For this example a 10% reduction in EI is made at element #4 and a 15% reduction in EI is made at element #6. The natural frequencies of the undamaged (FE/Analytic) and simulated damaged (Test) models are shown in Table 4.2. Using the baseline system, two solutions seemed to localize the error. The first was the solution using modes 1 to 5. This solution is plotted in Figure 4.2(a). The second was the solution using modes 1 to 4 and this solution is shown in Figure 4.2(b). However, it was known from the start that the damage was located at elements 4 and 6.

Mode Number	FE/Analytic (Hz)	Test (Hz)	% Error
1	31.30	30.04	4.0
2	86.30	85.16	1.3
3	169.29	167.19	1.2
4	280.26	276.22	1.4
5	419.78	413.91	1.4
6	588.73	581.20	1.3
7	788.17	778.01	1.3
8	1018.0	1004.5	1.3
9	1264.6	1246.0	1.5
10	1679.4	1656.7	1.4

**Table 4.2** Example 2: FE and Test Frequencies and Error

In a standard test, there would be no prior knowledge of damage and the results using the baseline system would appear to be accurate. This misleading information would not help in the updating of the FE model. Using the ABC system and modes 1 through 8, the solution shown in Figure 4.2(c) is obtained and proves to be a reliable solution based on the known error.



**Figure 4.2** Damage Detection – Example 2  
(a) & (b) Baseline (c) ABC system

### 3. Damage Detection: Example 3

For Example 3, two elements at opposite ends of the beam were chosen. A 15% reduction in EI was made at element #1 and a 20% reduction in EI was made at element #10. The natural frequencies of the undamaged and simulated damaged models are shown in Table 4.3. Using the baseline system, the only solution that provided any

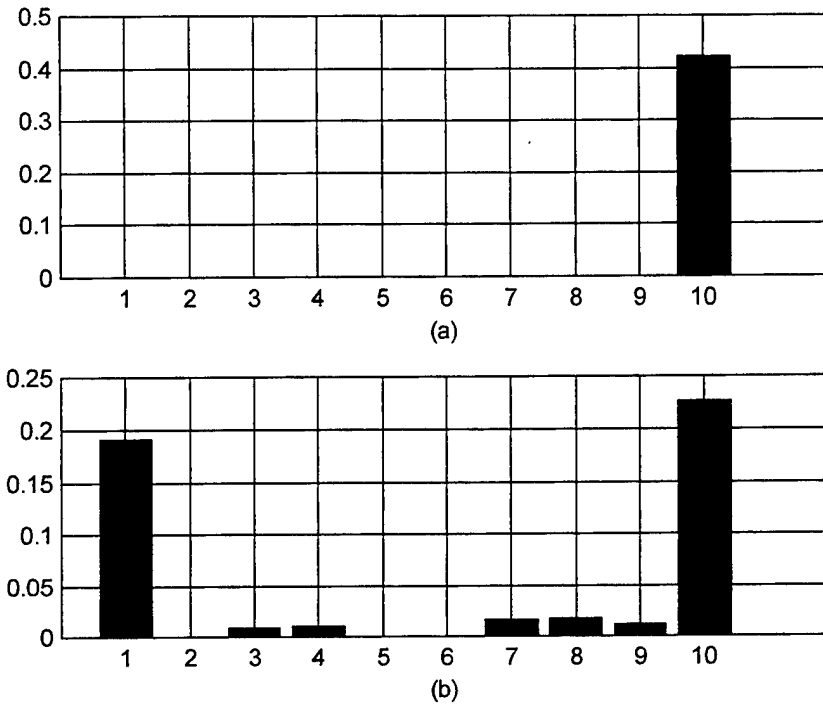
Mode Number	FE/Analytic (Hz)	Test (Hz)	% Error
1	31.30	31.29	0.0
2	86.30	86.21	0.1
3	169.29	168.78	0.3
4	280.26	278.44	0.6
5	419.78	415.12	1.1
6	588.73	579.27	1.6
7	788.17	772.08	2.0
8	1018.0	994.76	2.3
9	1264.6	1240.1	1.9
10	1679.4	1646.6	2.0

**Table 4.3** Example 3: FE and Test Frequencies and Error

error localization was the solution using modes 1 to 5. This solution is shown in Figure 4.3(a). It is immediately obvious that this solution is incorrect, as it indicates an error of approximately 42% in element #10 and no damage in any other elements. Using the ABC system and modes 1 through 7, an improved solution is obtained and is shown in Figure 4.3(b).

### 4. Damage Detection: Example 4

For the fourth example three elements were chosen to have reduced EI values. A 10% reduction in EI is made at element #3, a 15% reduction at element #4, and a 20%



**Figure 4.3** Damage Detection – Example 3  
(a) Baseline (b) ABC system

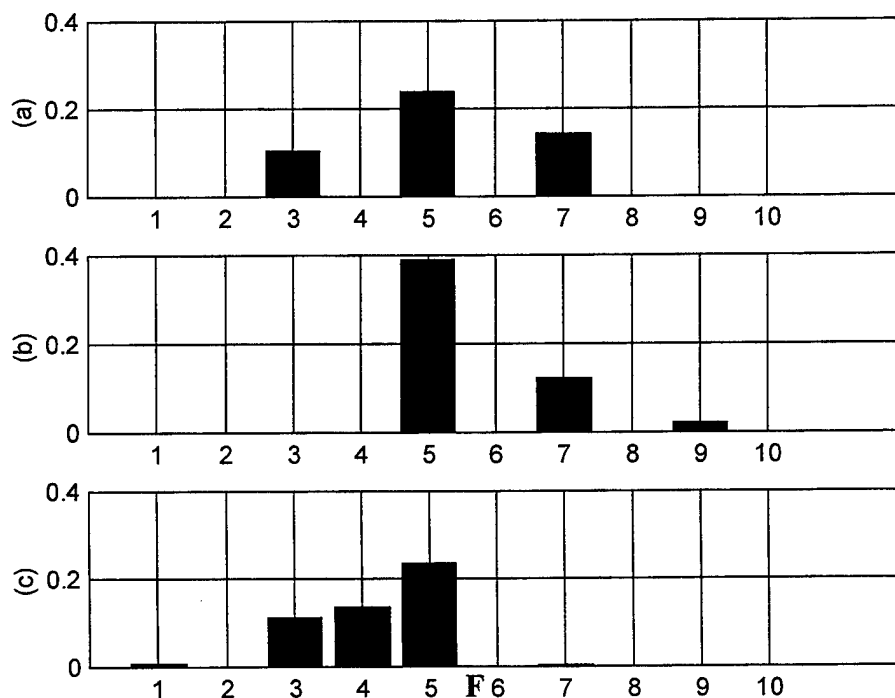
reduction at element #5. The natural frequencies of the undamaged and simulated damaged models are shown in Table 4.4. The best baseline system solution was obtained using modes 1 to 3 and is displayed in Figure 4.4(a). The solution localized error in three elements, but localized damage in element #7 instead of element #4. This is obviously incorrect. Using the ABC system and the same modes (1 to 3), the solution displayed in Figure 4.4(b) was obtained. This solution also is incorrect and misleading. By using the ABC system and modes 1 through 6, however, an improved and correct solution is obtained and is shown in Figure 4.4(c). It is important to note that in this fourth example, the ABC system also provided one misleading solution. Given the number of modes



Mode Number	FE/Analytic (Hz)	Test (Hz)	% Error
1	31.30	29.87	4.6
2	86.30	83.84	2.9
3	169.29	164.68	2.7
4	280.26	273.90	2.3
5	419.78	409.57	2.4
6	588.73	574.71	2.4
7	788.17	768.17	2.5
8	1018.0	993.73	2.4
9	1264.6	1231.7	2.6
10	1679.4	1637.8	2.5

**Table 4.4** Example 4: FE and Test Frequencies and Error

used, the solution using the first six modes would be considered more reliable even if the damage had not been known a priori due to its improved conditioning.

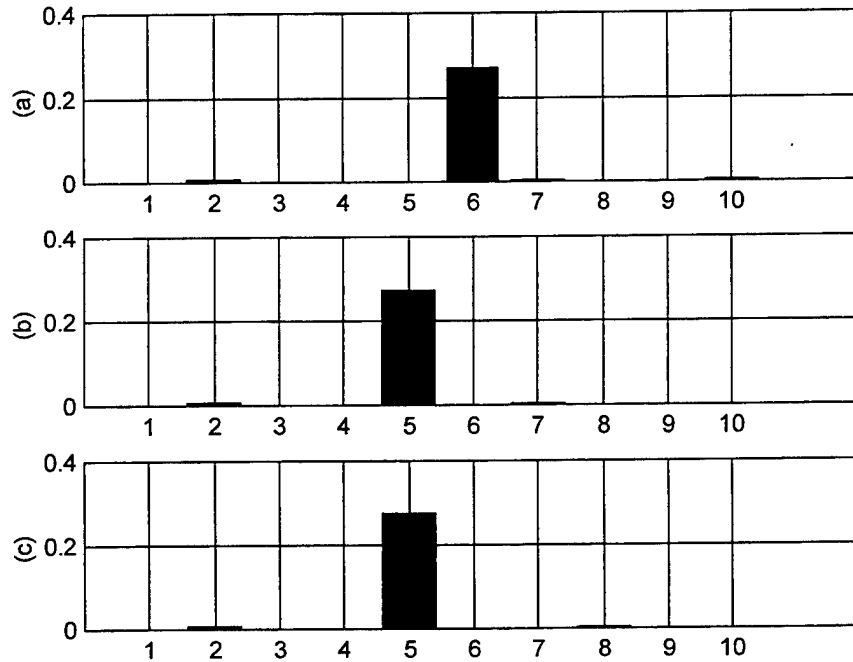


**Figure 4.4** Damage Detection – Example 4  
(a) & (b) Baseline (c) ABC system

For the final two examples, a random number generator is used to place errors in 1 to 2 elements of the model by reducing EI in each of those elements by up to 50%. Using this method, the damage location and magnitude is unknown and left to be determined by using both baseline and ABC system sensitivities. Examining the element EI matrix from the MATLAB code checks the results determined by the sensitivities.

### 5. Damage Detection: Example 5

The damage detection results obtained by using the maximum number of modes possible in the baseline system , five (rank=5), are shown in Figure 4.5 (a). It appears



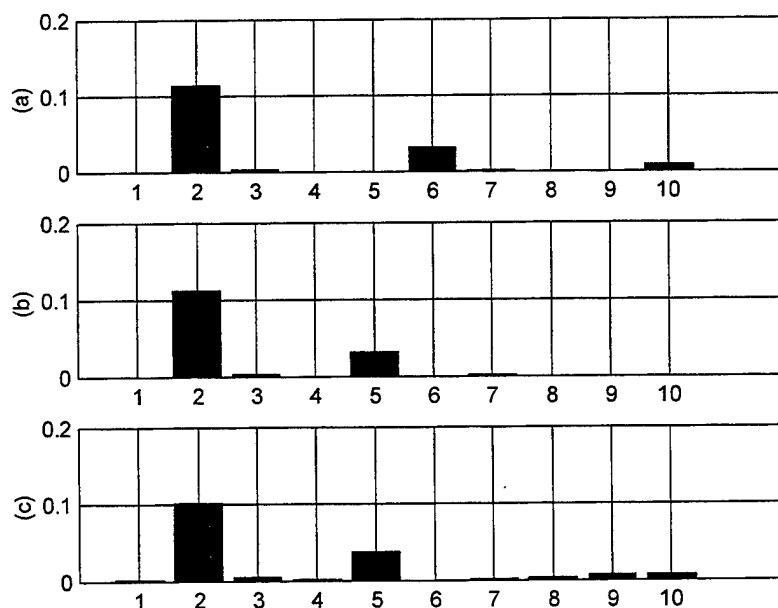
**Figure 4.5** Damage Detection – Example 5  
(a) & (b) Baseline (c) ABC system

from this plot that element #6 is the damaged element. However, when modes 1 through 4 are used in the baseline system (Figure 4.5(b)) it appears that #5 is the “damaged” element. By using modes 1 through 5 in the ABC system, Figure 4.5(c) was obtained and

verified that #5 was the damaged element. This was believed to be the best localization of error and the element EI matrix was checked for verification. The only damaged element was element #5 and the difference between damaged and undamaged was a 23 percent reduction, which corresponds to the values in Figure 4.5(b) and (c). Once again, the use of the ABC system provided reliable results when, without the ABC configuration, the baseline information alone may have been misleading.

## 6. Damage Detection: Example 6

Once again, the results using the first five modes with baseline sensitivity are displayed in Figure 4.6(a). Using this solution, it appears that the damage is located in elements #2 and #6. With the first four modes, the solution shows damage in elements #2 and #5 as shown in Figure 4.6(b). Figure 4.6(c) shows the results obtained when using all 10 modes and the ABC system sensitivity. It confirms that elements #2 and #5 are the



**Figure 4.6** Damage Detection – Example 6  
(a) & (b) Baseline (c) ABC system

damaged elements. Checking the EI element matrix, it was confirmed that element #2 had a 10% reduction in EI and element #5 had a 3% reduction in EI. These values correspond to those seen in Figure 4.6(b) and (c). In summary, all 6 examples have provided valuable information concerning damage detection. Many times, a reasonable solution can be obtained using baseline sensitivities. However, with this always seems to be some misleading information when the baseline solutions are used. The ABC system will most always provide the proper solution, but is not perfect either. When the ABC system is used in conjunction with the baseline system, improved and accurate solutions are obtained. Nowhere is this clearer than in the previous two examples.

## V. EXPERIMENTAL APPLICATION

So far, the use of the ABC configuration to determine natural frequencies of a system under a variety of boundary conditions has been demonstrated using MATLAB and computer generated finite elements. To fully examine this process, a laboratory experiment was constructed.

### A. EQUIPMENT SETUP

A mild steel beam, 60.75 inches in length, 1.5 inches wide and 0.5 inches thick, was suspended from the overhead by monofilament fishing line in order to represent a free-free beam with two rigid body modes. The beam was instrumented with 5 *PCB Piezotronics Inc.* accelerometers, each 15.25 inches apart. Each of the accelerometers was wired into a 5 kHz *Zonic System 7000* front end digital signal processor (DSP) consisting of 2 digital chassis with 16 channels each and 1 HP workstation. An excitation was applied successively to the opposite side of each of the accelerometers by a 25 lb peak force shaker and lateral excitation stand Model 2050A, manufactured by *The Modal Shop, Inc.* A *MB Dynamics Model SS250VCF* amplifier set to constant voltage powered the shaker. The shaker was operated at 0.25 volts (rms). The applied force was wired into the DSP as Channel 1, and the 5 accelerometers as Channels 2 through 6, respectively. The *Zonic* system was connected to the *I-DEAS* Master Series 5 software operated in the Signal Processing mode. The laboratory setup is shown in Figure 5.1.

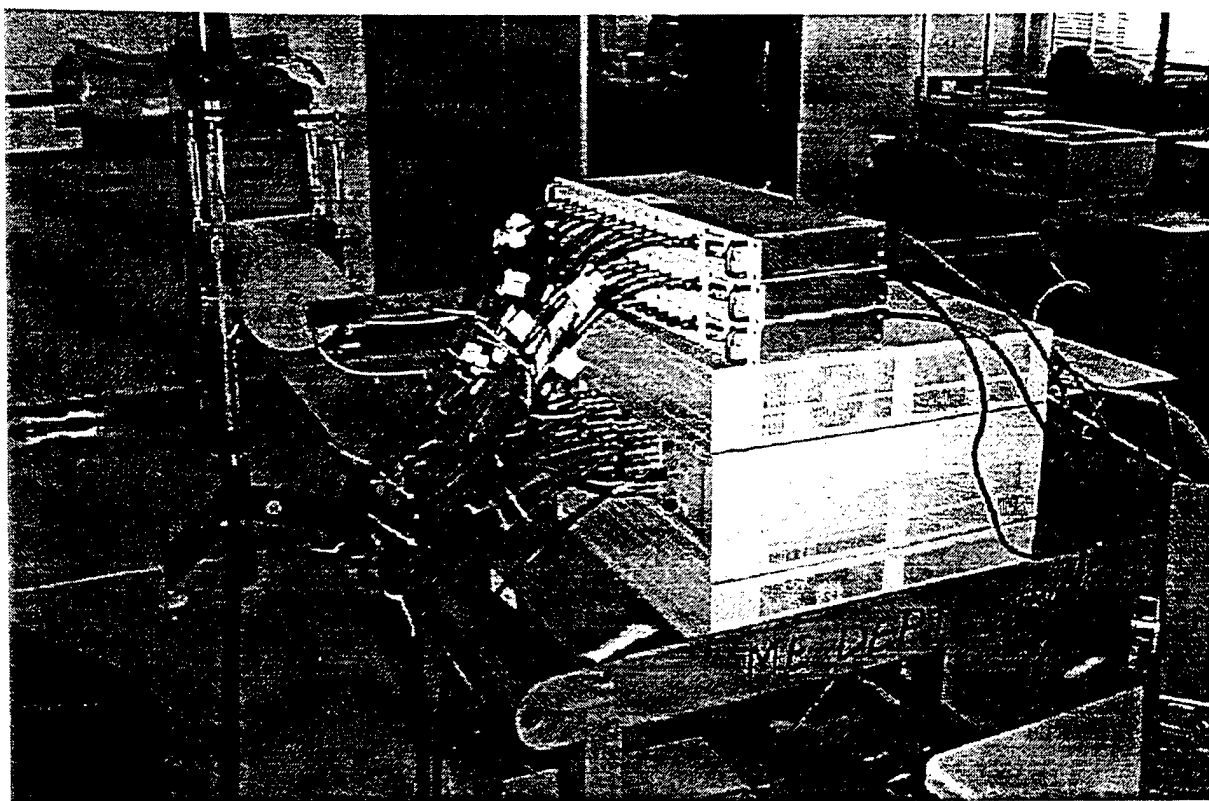
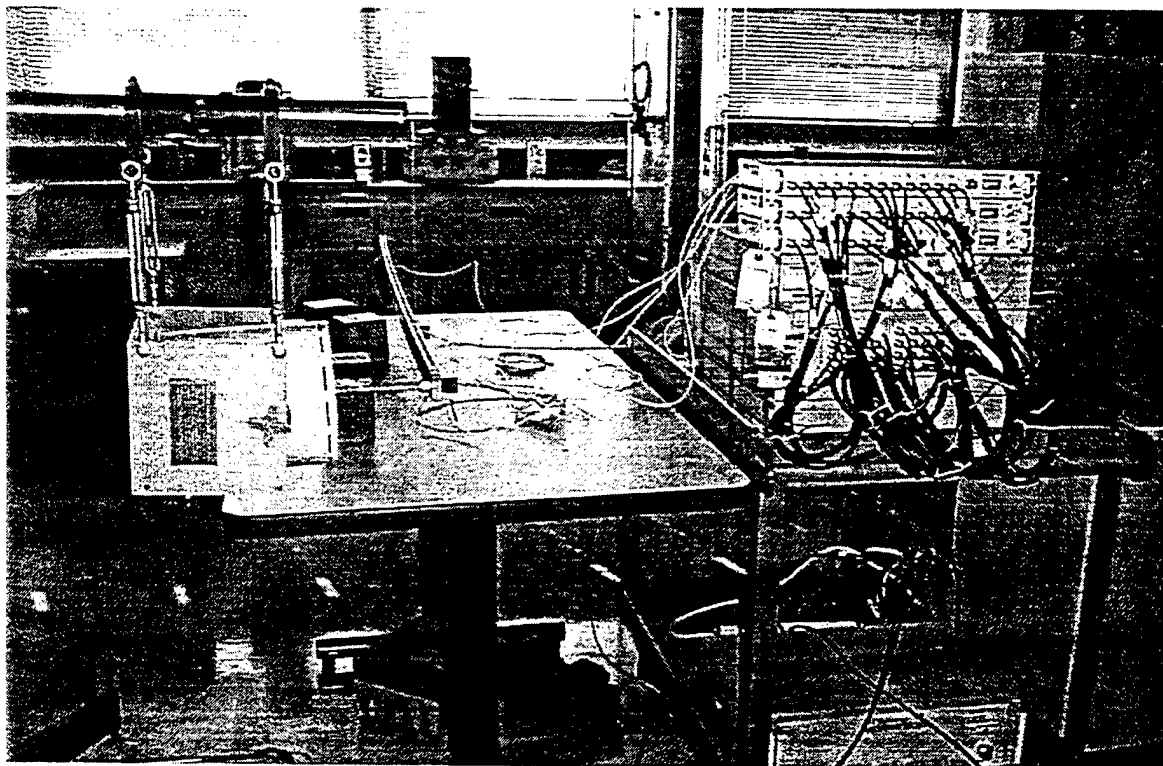


Figure 5.1 Laboratory Setup

## **B. DATA COLLECTION**

Using *I-DEAS*, FRFs were collected when the force was applied at each node. For each reference (i.e. for the force applied at each node), the response at each node was measured. *I-DEAS* was configured to measure 1601 spectral lines over frequencies ranging from 0 -2000 Hz. Each shaker provided a 1601 by 5 matrix containing the complex amplitudes every 1.25 Hz for a given excitation. The combined result provided a 1601 by 25 matrix for the entire system. This matrix was exported to MATLAB in order to compute the o-set frequencies of the system.

## **C. O-SET FREQUENCIES USING EXPERIMENTAL DATA**

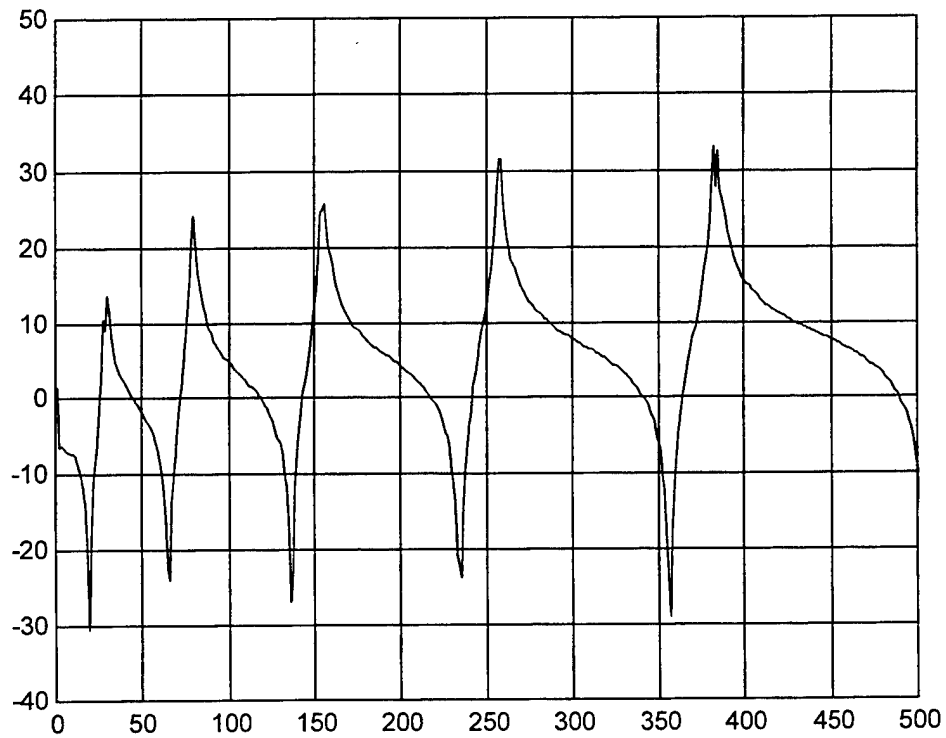
In order to establish a comparison and validate the results obtained in the laboratory, a finite element model of the beam was created. The FE model contained the given dimensions of the beam and a Modulus of Elasticity (E) of  $27.3 \times 10^6$  lbf/in<sup>2</sup>, corresponding to mild steel. This E value was chosen in order to bring the baseline model frequencies closer to the test frequencies. It was composed of 64 elements with 65 nodes (2 DOF per node for 130 DOF total). Nodes 1, 17, 33, 49, and 65 corresponded to transducer locations 1 through 5, respectively, on the real beam.

The theories presented earlier and demonstrated through computer simulation were applied to the experimental data. By using ABC configurations, the frequencies for the beam under 30 different boundary conditions can be found from the baseline (free-free) information without having to reconfigure the experimental setup. Table 5.1 contains the first ten natural frequencies (in Hz) of the system computed from the test and compares them to the natural frequencies determined by the FE model. Figure 5.2

contains a plot of the 1,1 element of the FRF ( $H(\Omega)$ ) matrix and displays the first five modes.

FE/Analytic (Hz)	Test (Hz)	% Difference
29.1	29.0	0.3
80.3	79.5	1.0
157.4	155.3	1.3
260.2	257.8	0.9
388.7	388.3	0.1
542.8	534.0	1.6
722.7	709.0	1.9
928.3	912.9	1.7
1159.6	1104.1	4.8
1416.6	1393.5	1.6

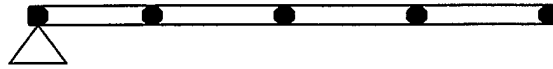
**Table 5.1** FE & Test frequencies for the free-free beam



**Figure 5.2**  $H_{11}(\Omega)$  for free-free beam



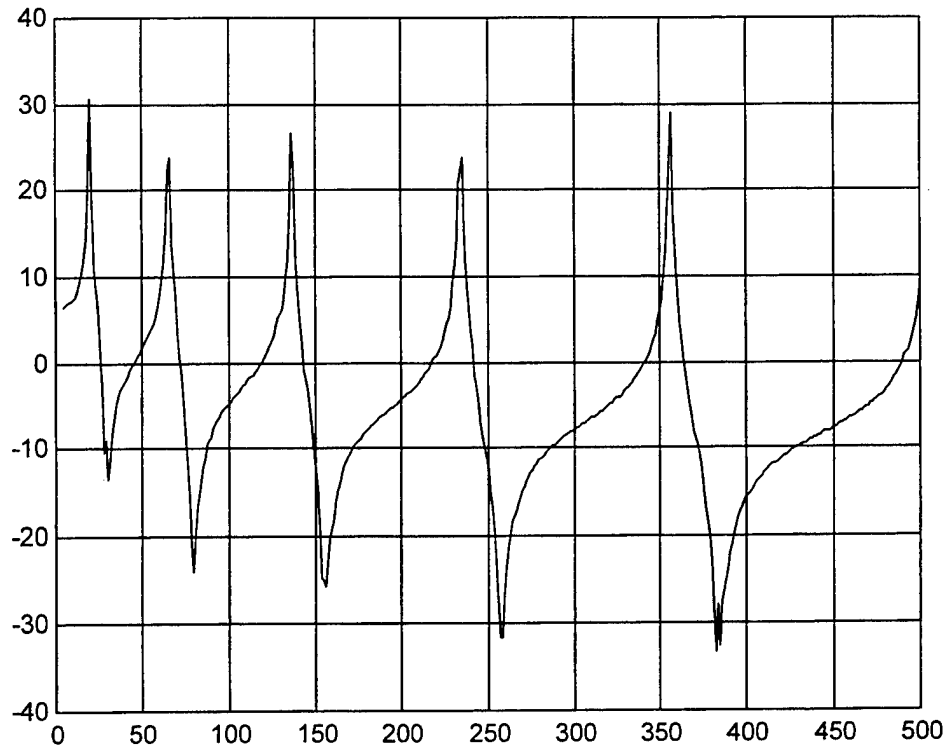
The first o-set system computed was the system with node 1 restrained to ground, corresponding to the system shown in Figure 5.3. The FRF matrix was inverted to find the impedance matrix for the a-set ( $H_{aa}^{-1}(\Omega)$ ) and the peaks of the impedance matrix correspond directly to the natural frequencies of the system with node 1 restrained to ground. These o-set frequencies were compared to the o-set frequencies determined by the FE model and are shown in Table 5.2. A plot of  $H_{aa}^{-1}(\Omega)$  showing the first five frequencies is seen in Figure 5.4.



**Figure 5.3** ABC configuration: Node 1 restrained to ground

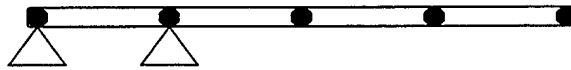
FE/Analytic (Hz)	Test (Hz)	% Difference
20.0	20.0	0.0
65.0	65.6	0.9
135.7	136.0	0.7
232.1	235.0	1.2
354.1	356.2	0.6
501.9	503.8	0.4
675.3	669.6	0.8
874.5	873.3	0.1
1099.3	1080.7	1.7
1349.9	1354.1	0.3

**Table 5.2** FE & Test Frequencies for Node 1 restrained to ground



**Figure 5.4**  $H_{aa}^{-1}(\Omega)$  for Node 1 restrained to ground

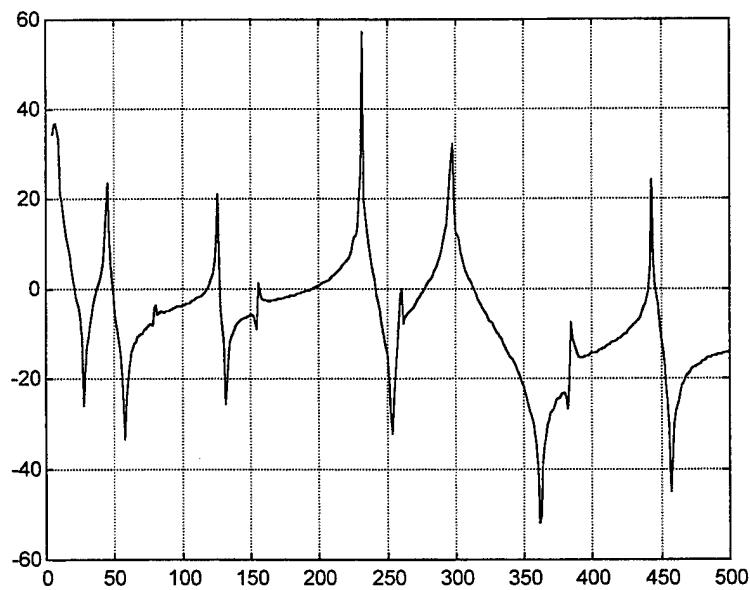
In both cases, the test frequencies correspond nicely with the frequencies predicted by the FE model. This validates the theories and expectations described in the earlier chapters. With only free-free data available, the natural frequencies of the system under any boundary configuration can be found. The number of transducers (i.e. measurement locations) limits the number of systems for which frequencies are available and in the given example, up to 30 system frequencies can be found. Six more ABC systems were computed and the results for the first ten modes are tabulated in Tables 5.3 - 5.8. The system configurations and  $H_{aa}^{-1}(\Omega)$  plots are shown in Figures 5.5 - 5.16.



**Figure 5.5** ABC Configuration: Nodes 1 & 2 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
6.8	7.2	5.9
44.3	44.7	0.9
125.7	125.9	0.2
228.9	231.9	1.3
293.4	297.0	1.3
443.1	443.9	0.2
652.7	648.4	0.7
868.4	867.9	0.1
990.2	980.5	1.0
1252.2	1253.1	0.1

**Table 5.3** FE & Test Frequencies for Nodes 1,2 restrained to ground



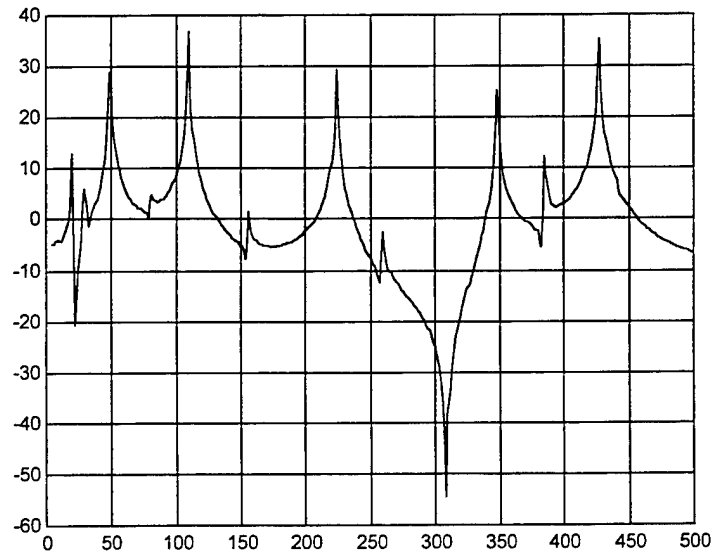
**Figure 5.6**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,2 restrained to ground



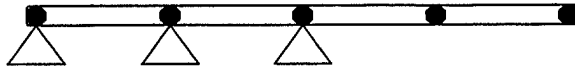
**Figure 5.7** ABC Configuration: Nodes 1,4 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
20.0	20.0	0.0
49.4	49.1	0.4
108.0	110.0	1.8
219.5	223.9	2.0
349.6	347.6	0.6
427.6	427.2	0.1
606.6	618.8	2.0
848.7	856.0	0.9
1092.5	1045.2	4.3
1228.7	1229.7	0.1

**Table 5.4** FE & Test Frequencies for Nodes 1,4 restrained to ground



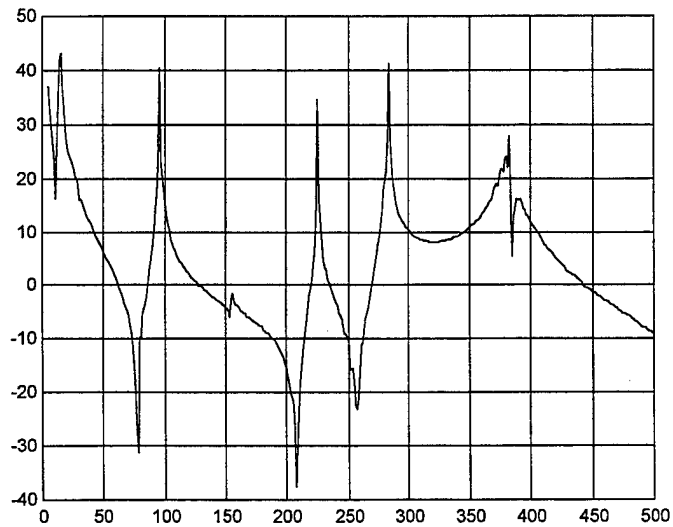
**Figure 5.8**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,4 restrained to ground



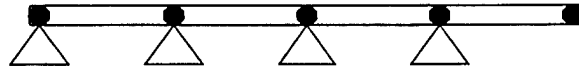
**Figure 5.9** ABC Configuration: Nodes 1,2,3 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
14.5	15.0	3.4
96.5	96.8	0.3
220.8	225.1	1.9
282.2	283.4	0.4
375.0	381.1	1.6
588.3	537.4	*8.7
853.2	858.5	0.6
970.9	957.7	1.4
1133.3	1126.9	0.6
1489.3	1463.6	1.7

**Table 5.5** FE & Test Frequencies for Nodes 1,2,3 restrained to ground



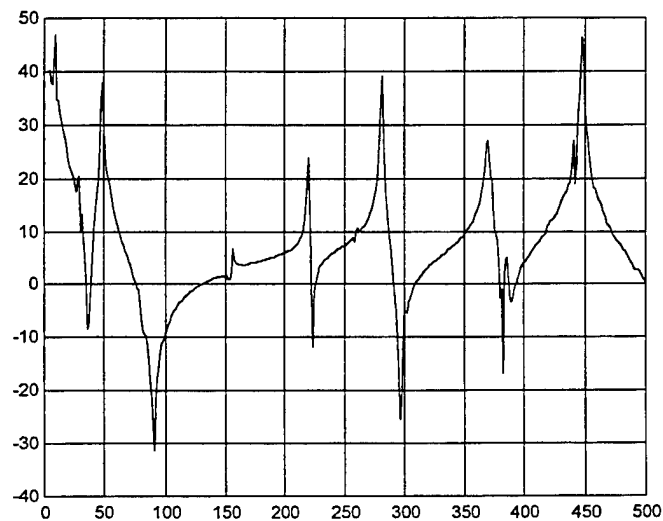
**Figure 5.10**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,2,3 restrained to ground



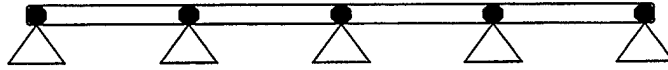
**Figure 5.11** ABC Configuration: Nodes 1,2,3,4 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
49.3	48.3	2.0
215.3	220.0	2.2
277.4	281.2	1.2
367.5	369.5	0.5
448.2	448.1	0.0
841.6	848.2	0.8
961.6	956.5	0.5
1123.0	1123.2	0.0
1260.9	1258.4	0.2
1879.2	1895.6	0.9

**Table 5.6** FE & Test Frequencies for Nodes 1,2,3,4 restrained to ground



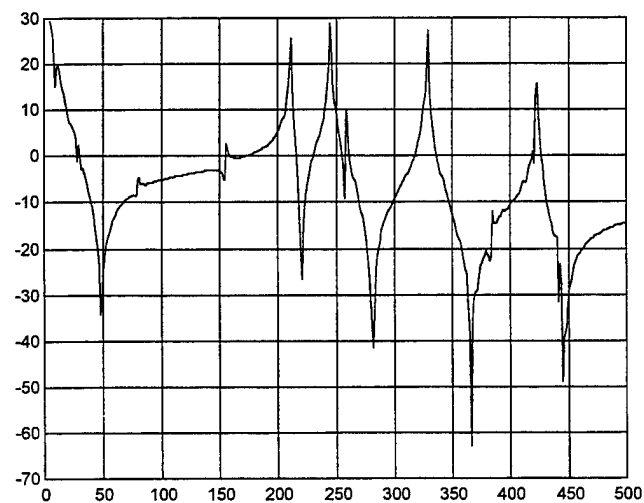
**Figure 5.12**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,2,3,4 restrained to ground



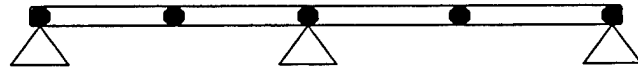
**Figure 5.13** ABC Configuration: Nodes 1,2,3,4,5 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
205.6	210.6	2.4
239.8	245.2	2.3
321.1	328.2	2.2
415.0	422.1	1.7
822.3	831.1	1.1
892.4	894.8	0.3
1040.7	1043.7	0.3
1200.4	1214.1	1.1
1850.3	1869.1	1.0
1954.5	1952.8	0.1

**Table 5.7** FE & Test Frequencies for Nodes 1,2,3,4,5 restrained to ground



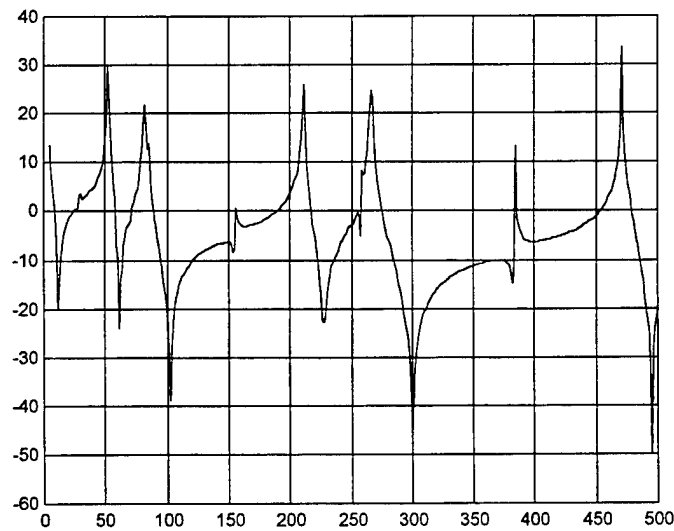
**Figure 5.14**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,2,3,4,5 restrained to ground



**Figure 5.15** ABC Configuration: Nodes 1,3,5 restrained to ground

FE/Analytic (Hz)	Test (Hz)	% Difference
51.4	52.2	1.6
80.3	82.4	2.6
205.6	210.7	2.5
260.2	266.5	2.4
462.5	471.1	1.9
542.8	554.8	2.2
822.3	831.8	1.2
928.3	940.9	1.4
1284.8	1306.8	1.7
1416.6	1428.6	0.8

**Figure 5.8** FE & Test Frequencies for Nodes 1,3,5 restrained to ground



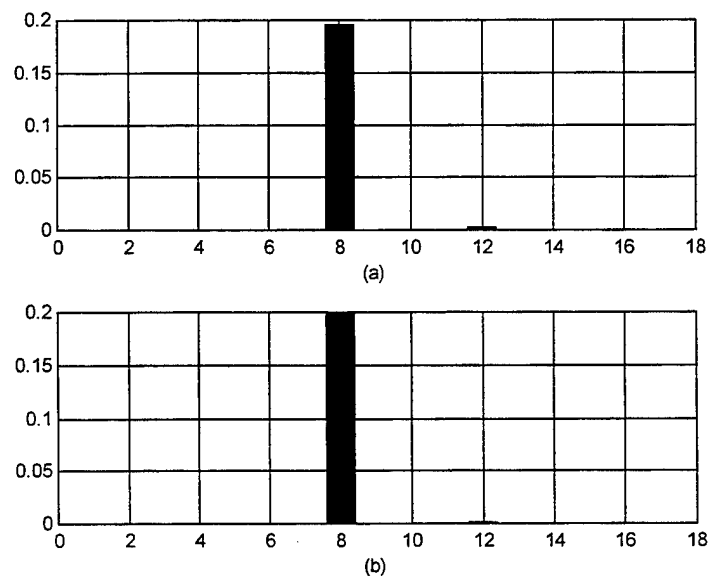
**Figure 5.16**  $H_{aa}^{-1}(\Omega)$  for Nodes 1,2,3,4,5 restrained to ground



#### D. DAMAGE DETECTION USING EXPERIMENTAL DATA

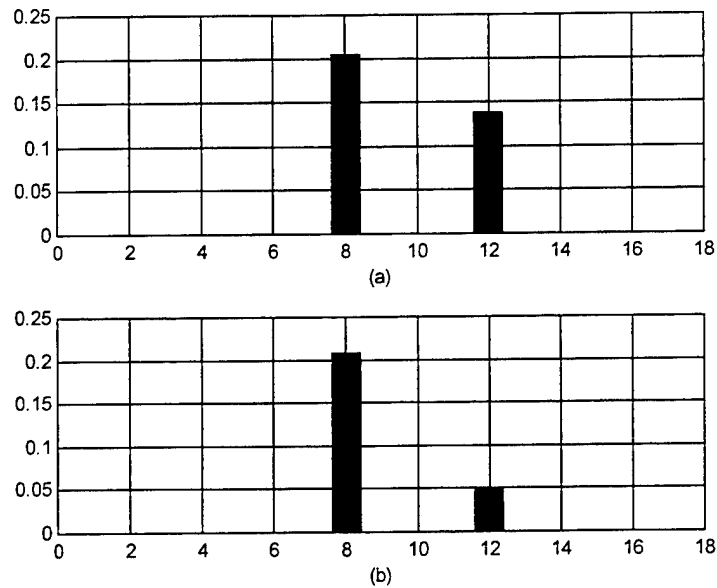
The next logical step to take is to try to localize error in a FE model by using the experimental data. In Chapter IV, sensitivity-base damage detection was used to localize error in a "damaged" FE model by using another FE model with no damage. We now have a set of data for a real beam that has no damage. The experimental data is now needed in order to find the error in a FE model of the same system. To perform these calculations, an FE model was again created, this time of 16 elements. Fewer elements were used in order to decrease computational time; the actual change in frequencies from those determined using 64 elements was negligible. A 20% reduction in EI was made at element #8 in the FE model. A model of the free-free beam (baseline) was created as well as a model of the ABC system with nodes 1 and 4 restrained to ground.

To check the method, FE frequencies were substituted for the test frequencies in both the baseline system and the ABC system. The results are shown in Figure 5.17.



**Figure 5.17** Error localization using FE frequencies  
(a) Baseline (b) ABC system

The error was localized using the first two modes of the baseline system (Figure 5.17(a)) and then was confirmed using the first two modes of the ABC system (Figure 5.17(b)). This validated the method and now the FE frequencies were replaced by the actual test frequencies for the system. These results are shown in Figure 5.18.



**Figure 5.18** Error localization using test frequencies  
(a) Baseline (b) ABC system

The first two modes were again used for both the baseline system (a) and the ABC system (b) since these were the modes that resolved the error when the FE frequencies were used. This time, however, the baseline system localized a 20% error in element #8 as well as a 13% error in element #12. Clearly this is incorrect. The ABC system provided better results, but also indicated error in element #12 at 5%. The differences in the frequencies between the test data and the FE data, however slight, are

the sources of the discrepancies. In the first two modes of the free-free beam, the test frequencies differed from the FE frequencies by 0.3% and 1.0%, respectively (see Table 5.1). For the ABC system's first two modes, the test frequencies differed by 0.0% and 0.4%, respectively (see Table 5.4). Therefore, in order to detect and localize errors in FE models with test information, it is imperative that the error between test and FE frequencies be minimized. Suggested approaches on how to achieve this are proposed in the Recommendations.



## **VI. CONCLUSIONS AND RECOMMENDATIONS**

The main objective of this thesis was to examine a method in which the natural frequencies of a system under a variety of boundary conditions could be found by using only one experimental database. Previously, experiments had to be reconfigured in order to find more than the baseline frequencies. This results in more time and inevitably, an increase in cost for structural analysis. Secondly, a method by which model error detection and localization could be improved was investigated.

### **A. CONCLUSIONS**

The following conclusions can be drawn from the analyses presented:

1. The natural frequencies of a structure under various boundary conditions are available from a baseline FRF matrix. Inverting a spatially incomplete FRF at each frequency results in an impedance spectrum whose peaks correspond to the natural frequencies of the structure restrained to ground at measure coordinates.
2. With both the baseline and ABC frequencies available, structural sensitivities can be generated from both the baseline and from the system with boundary conditions artificially applied corresponding to the FRF matrix (a-set). The addition of the ABC data has proven to be valuable in conjunction with the baseline information in localizing error and/or damage.

## **B. RECOMMENDATIONS**

The following are areas of recommended improvement to continue with this field of study. The first two are recommendations for improving sensitivity-based error localization when using test data.

1. Curve fit the FRF in *I-DEAS* prior to exporting the files into MATLAB.
2. Once in MATLAB, curve fit the impedance spectra near the ABC frequency to identify the ABC natural frequencies.
3. Continue with the use of ABC systems on more advanced structures to further this study.

## APPENDIX MATLAB CODE

### A. MAIN PROGRAMS

#### Build2Beams.m

```
% This program assembles the mass and stiffness matrices for 2 beams, with user
% defined boundary conditions referred to as "BeamA" (analysis) and "BeamX"
% (experimental). The program can be run in both "build" and "read" modes. In the
% "build mode," the user provides baseline data for BeamA, assumed to be a
% homogeneous, uniform beam. Data provided:
%
% (1) Beam length
% (2) Number of elements
% (3) Nominal EI
% (4) Nominal cross-sectional area
% (5) Nominal weight density
%
% The program then prompts the user for instructions on how to modify "BeamA"
% data to arrive at "BeamX" data. The user can modify element EI values and the
% modification can be applied to either a single element, or range of elements. The
% beam definition data is saved in a binary (.mat) file "beamdata" at the end of
% execution.
%
% In the "read" mode, the program will load the beam data written at the last
% execution of the program and will then allow the user to modify the data for
% BeamX. By answering "y" to this prompt, the program will copy BeamA data into
% BeamX data, and then prompt the user (identically as described above) for data on
% how to adjust the BeamA data in order to arrive at the BeamX data.
%
% The program calls the following support programs:
%
% BeamA_Prompt.m      -- Prompts User for BeamA nominal beam data
%                      ("build" mode only)
% BeamX_Prompt.m      -- Prompts User for BeamX modification beam data
% BoundaryConditions.m -- Prompts user for B.C.'s and applies them.
% Assemble2Beams.m    -- Called by Build2Beams, builds [ka] [ma] [kx] [mx],
%                      plots freqs.
%                      Saves data to "beamdata.mat"
% clearspace.m        -- Clears all unnecessary variables from workspace.
%
%
% Written by J.H. Gordis
% Modified by C.P. DeGregory
```

```

disp(' ');
disp(' Build2Beams execution...');

input_flag = 'b';
input_flag = input(' "B"uild new beams or "R"ead from file ? (b/r) ','s');

% Build_or_read if loop:

% *****
if input_flag == 'b'; % (IF#1) Prompt user for beam definition data
% *****

    disp(' Building 2 beams from scratch...')

    BeamA_Prompt; % Prompt for BeamA Data: run prompt script
    BeamX_Prompt; % Prompt for BeamX Modification Data: run prompt script
    Assemble2Beams; % Run script to assemble mass and stiffness.
    BoundaryConditions; % Prompt for, and apply boundary conditions

%     *** Save Defining Parameters for Beams ***

    disp(' ...saving beam data to file')
    save beamdata connect ndof element_length element_EI element_mass mass_diag
    restraint_switch free_dof_set...
    num_elements num_rbm lama lamx
    %save beamdata phia phix

% *****
elseif input_flag == 'r'; % (IF#1) file read for beam definition data
%*****

    disp(' Loading 2 beams from ascii file "beamdata"...')
    load beamdata

% Prompt user to either build two beams from data in file, or modify beamx data:
    modify_beamx = 'n';
    modify_beamx = input(' Modify beamX data (y/n)? ','s');

    if modify_beamx == 'y'; % (IF#2)
        disp(' Modify BeamX...')
        reset_beamx = 'y';
        reset_beamx = input(' Reset BeamX properties to BeamA properties? ...
                            (y/n) ','s');
        if reset_beamx == 'y'; % Copy beamA data to beamX data:

```



```

        element_EI(:,2)=element_EI(:,1);
        element_mass(:,2)=element_mass(:,1);
%      ...Display BeamA properties to user:
        sprintf('Number of elements: %3i',num_elements)
    end; % end reset_beamx

BeamX_Prompt;      % Prompt for BeamX Modification Data: run script

% End modify_beam if block (IF#2)
end

Assemble2Beams;    % Run script to assemble mass and stiffness.
BoundaryConditions; % Apply boundary conditions (that already exist)

%      *** Save Defining Parameters for Beams ***

disp(' ...saving beam data to file')
save beamdata num_elements connect ndof element_length element_EI ...
        element_mass mass_diag restraint_switch free_dof_set...
        lama phia lamx phix num_elements connect ka ma kx mx num_rbm

% *****
end;      % End Build or read if loop IF#1
%*****

disp(' Build2Beams end.')
% _____

```

## Beam\_Sensitivity.m

```
% This program calculates the stiffness and mass matrix partials by finite
% differences. That is, for example, the [k] matrix is assembled twice, once for the
% nominal beam parameters, and a second matrix is assembled based on a small
% perturbation of element EI.
%
% This program makes use of the beam data created by the program
% "Build2Beams.m," reads the data file BeamData and resets BeamX EI data to be
% the same as BeamA data.
%
% The user is prompted for element label numbers to be used as design variables by
% the script "BeamSens_Prompt" which stores this element label data in the matrix
% "EI_el_lbl."
%
% EI_el_lbl: This array is (max_num_dv,num_elements), where max_num_dv is
% the maximum number of design variables allowed, and each design variable may
% include from 1 to "num_elements" elements. Therefore, the sets of elements
% included in two design variables may share common elements.
%
% Based on the number of non-zero rows in the array "EI_el_lbl," matrix partials
% are calculated for each non-zero row. For example, to start, the first row of
% EI_el_lbl is read, and a small perturbation in element EI is applied simultaneously
% to the elements listed in this row.
%
% The script generates a matrix containing mode frequency sensitivities:
%
% [sens_EI]: num_modes * size(EI_el_lbl,1)
%
%
% Written by J.H. Gordis
% Modified by C.P. DeGregory
%
%*****
% Start code:
% ~~~~~

format long;

disp(' ');
disp(' BeamSensitivity execution...');disp(' ');
disp(' Loading 2 beams from ascii file "beamdata"...');disp(' ');
load beamdata
num_elements = num_elements
```

```
clear num_mass_dv num_EI_dv define_M_K_dv EI_el_lbl mass_el_lbl
clear sens_EI sens_mass
```

```
BeamSens_Prompt; % Prompt for DV element label definitions
```

```
% *****
% ***** INITIALIZATION *****
% *****
```

```
mass_change = 1; % Percentage mass change for sensitivity calculation.
EI_change = 1; % Percentage EI change for sensitivity calculation.
element_mass_orig = element_mass; % Copy properties over to retain them.
element_EI_orig = element_EI; %
```

```
icnt_dv = 0; % Initialize number of DV's to zero. Will count up all DV's
```

```
% Prompt for number of mode frequencies to generate sensitivities for:
```

```
% ~~~~~
```

```
num_modes = input(' Enter number of elastic modes for sensitivity calculations>> ');
```

```
start_mode = num_rbm + 1; % Skip the rigid body modes.
```

```
% *****
% ***** EI SENSITIVITY CALCULATION LOOP *****
% *****
```

```
num_EI_dv = size(EI_el_lbl,1);
```

```
for icnt_dv = 1:num_EI_dv; % Repeat all of this for all EI DV's defined.
```

```
    % Resetting BeamX properties to BeamA properties...');disp(' ');
    element_EI(:,2) = element_EI(:,1);
```

```
    % Get element labels for current DV:
    lbls = EI_el_lbl(icnt_dv,find(EI_el_lbl(icnt_dv,:)));
```

```
    % Perturb EI props for all elements in "lbls":
```

```
    element_EI(lbls,2) = element_EI(lbls,2) * (1 + (EI_change/100));
```

```
    Assemble2Beams; % Run script to assemble baseline and perturbed beams.
```

```

BoundaryConditions; % Apply boundary conditions.

[lam_base,phi_base]=fmodes2(ka,ma);      % Get modes of baseline beam
% Form EI derivative matrices:
k_delta = (kx - ka)/(EI_change/100); % in %/100

% Mode freq sens loop:
end_mode = start_mode + (num_modes - 1); % Get requested number of modes.
row_num = 0;
for icnt_modes = start_mode:end_mode;
    row_num = row_num + 1;
    sens_EI(row_num,icnt_dv) = phi_base(:,icnt_modes)' * ...
        k_delta * phi_base(:,icnt_modes);
end;

end;

% End "for icnt_dv" outer loop for sensitivity calculations
% Run script to clear out variables:

clearspace

% Copy element EI and mass properties back into arrays:
element_EI = element_EI_orig;
element_mass = element_mass_orig;

save beamdata element_EI element_mass sens_EI
% Save all variables to beamdata (essentially does an append to existing beamdata)

disp(' BeamSensitivity end.')
% _____

```

## Beamex.m

```
% This program calculates an FRF matrix and the inverse of an FRF matrix in order
% to identify the Artificial Boundary Condition frequencies. It loads the binary
% (.mat) file "Beamdata" and determines the natural frequencies of the system from
% the stiffness and mass matrices by calling the function "fmodes." The program
% also calls the function "fOset_from_Aset" to compute the o-set from a user
% defined a-set, as well as "fmodes" again, this time to solve for the o-set natural
% frequencies (which correspond to the ABC frequencies). The FRF is
% calculated as the inverse of the Impedance ( $H = Z^{-1}$ , where  $Z = k - (2m)$ ). Both
% the FRF and inverse FRF (Impedance) plots are displayed over the frequency
% range 0 – 800 Hz.
```

```
disp('-----')
```

```
load beamdata
```

```
[lam,phi] = fmodes(kx,mx);
freq = sqrt(lam)/2/pi;
```

```
% User is asked to input a-set
aset = input('Enter a-set (vector form): ');
```

```
[oset] = fOset_from_Aset(length(mx),aset);
```

```
ko = kx(oset,oset);
mo = mx(oset,oset);
```

```
disp('A-Set Defined as:')
aset
disp('O-Set Frequencies')
[lamo,phio] = fmodes(ko,mo);
freqo = sqrt(lamo)/2/pi;
icnt = 0;
freq = [1 : 1 : 5030];
freqp = ifreq / 2 /pi;
```

```
for ifreq = ifreq;
```

```
    icnt = icnt + 1;
    h = inv(kx + .001*sqrt(-1) * kx - ifreq^2 * mx);
    hx = h(aset,aset);
```

```

hxp11(icnt) = hx(1,1);
%      hxp22(icnt) = hx(2,2);
%      hxp33(icnt) = hx(3,3);
%      hxp44(icnt) = hx(4,4);
%      hxp55(icnt) = hx(5,5);
%      hxp66(icnt) = hx(6,6);

zabc = inv(hx);
zabc11(icnt) = zabc(1,1);
%      zabc22(icnt) = zabc(2,2);
%      zabc33(icnt) = zabc(3,3);
%      zabc44(icnt) = zabc(4,4);
%      zabc55(icnt) = zabc(5,5);
%      zabc66(icnt) = zabc(6,6);

end

%      Plot FRF and Impedance
clf
figure(1)
plot(freqp,log10(hxp11));grid
xlabel('Frequency (Hz)')
axis([0,800,-7,3])

figure(gcf+1)
plot(freqp,log10(zabc11));grid
xlabel('Frequency (Hz)')
axis([0,800,-1,7])

save hdata freqp zabc11

```

## Solveos.m

```
% This program solves for the vector {(EI)} following user guidelines. It first loads
% data which was saved after two runs of the "Build2Beams" and
% "Beam_Sensitivity" programs. The first run is for a baseline free-free beam with
% given element EI modifications. The analytical and experimental (simulated
% damaged) eigenvalues (lama & lamx) and the sensitivity matrix (sens_EI) are
% stored in beam0.mat. The second run is for a beam with user defined constraints
% at given measurement locations (a-set) and lama, lamx and sens_EI for this case
% are stored in beam1.mat.
%
%
% Written by J.H. Gordis
% Modified by C.P. DeGregory
```

```
load beam0
lama0=lama;
lamx0=lamx;
T0 = sens_EI;
clear lama lamx sens_EI
```

```
load beam1
lama1=lama;
lamx1=lamx;
T1 = sens_EI;
clear lama lamx sens_EI
```

```
rerun = 'y';
while rerun == 'y'
```

```
% Prompt for number of Beam mode frequencies:
% ~~~~~
% Note: for free-free structures, only elastic modes are in T matrix
```

```
num_modes0 = size(T0,1);
num_modes1 = size(T1,1);
```

```
sprintf('Total Beam0 mode sens. data avail: %3i', num_modes0)
sprintf('Total Beam1 mode sens. data avail: %3i', num_modes1)
```

```
disp(' '); disp(' For Beam0: ');
use_all_modes = input(' Use all Beam0 modes ? (y/n) >> ','s');
```

```
if use_all_modes == 'n';
```

```

        disp(' Enter (elastic) mode numbers to include...')
        disp(' Use MATLAB vector format> 1 3 5:7 9 ')
        use_modes0 = input(' >> ','s');
        use_modes0 = eval(['[',use_modes0,']']);
    else
        use_modes0 = 1:num_modes0;
    end;

    disp(' '); disp(' For Beam1: ');
    use_all_modes = input(' Use all Beam1 modes ? (y/n) >> ','s');

    if use_all_modes == 'n';
        disp(' Enter (elastic) mode numbers to include...')
        disp(' Use MATLAB vector format> 1 3 5:7 9 ')
        use_modes1 = input(' >> ','s');
        use_modes1 = eval(['[',use_modes1,']']);
    else
        use_modes1 = 1:num_modes1;
    end;

    % Prompt for number of DV's:
    % ~~~~~~

    num_DV0 = size(T0,2);
    num_DV1 = size(T1,2);

    sprintf('Total Beam0 DVs avail: %3i', num_DV0)
    sprintf('Total Beam1 DVs avail: %3i', num_DV1)

    % Prompt user for number of DVs to include:
    % ~~~~~~

    disp(' '); disp(' For Beam0: ');

    sprintf('Number of Beam0 EI DVs: %3i',num_DV0)
    use_all_dv = input(' Use all Beam0 DVs ? (y/n) >> ','s');
    if use_all_dv ~= 'y';
        disp(' Enter DV label(s) to include...')
        disp(' Use MATLAB vector format> 1 3 5:7 9 ')
        use_dv0 = input(' >> ','s');
        use_dv0 = eval(['[',use_dv0,']']);
    else
        use_dv0 = 1:num_DV0;
    end

```



```

disp(' '); disp(' For Beam1: ');
sprintf('Number of Beam1 EI DVs: %3i',num_DV1)
use_all_dv = input(' Use all Beam1 DVs ? (y/n) >> ','s');
if use_all_dv ~= 'y';
    disp(' Enter DV label(s) to include...')
    disp(' Use MATLAB vector format> 1 3 5:7 9 ')
    use_dv1 = input(' >> ','s');
    use_dv1 = eval(['[',use_dv1,']']);
else
    use_dv1 = 1:num_DV1;
end

% Assemble Composite Sensitivity Matrix:
% ~~~~~

format short

Tboth = [T0(use_modes0,use_dv0);T1(use_modes1,use_dv1)];

% Calculate Frequency Error Vector for the modes retained:
% ~~~~~

del_freq = [lamx0(use_modes0)-lama0(use_modes0);lamx1(use_modes1)-...
lama1(use_modes1)];

% Solve system for DV changes:
% ~~~~~

sprintf('Min Dim of T: %3i',min(size(Tboth)))
sprintf('Rank of T: %3i',rank(Tboth))
sprintf('Condition of T: %5.3e',cond(Tboth))

delta_dv = Tboth \ del_freq

bar(1:10,abs(delta_dv)),grid

rerun = input(' Rerun for different case ? (y/n) >> ','s');

end; % end rerun while loop

```

## Main.m

```
% This program is similar to the "beamex" program in that it computes FRF and
% Impedance matrices. This script, however, is for a spring-mass system. In
% particular, this program solves for the system shown in Figure 3.1. The stiffness
% and mass matrices are computed using the function "fspringmass" and the values
% for connectivity, mass specification and fixity. The values required for the system
% shown, in addition to the a-set (DOF #1), are contained in the script "testdata."
% The 1,1 elements of the FRF and Impedance matrices are plotted versus
% frequency.
```

```
clear
```

```
testdata
```

```
[k,m] = fspringmass(conn,mspec,fix);
[lam,phi] = fmodes(k,m);
```

```
[oset] = fOset_from_Aset(length(m),aset);
```

```
ko = k(oset,oset);
mo = m(oset,oset);
```

```
icnt = 0;
ifreq = [0:.01:4];
freqp = ifreq/2/pi;
h = zeros(size(k));
for ifreq = ifreq
    icnt = icnt + 1;
    h = inv(k - ifreq^2*m);
```

```
    hxp11(icnt) = h(1,1);
    %hxp22(icnt) = h(2,2);
    %hxp33(icnt) = h(3,3);
    %hxp44(icnt) = h(4,4);
    %hxp55(icnt) = h(5,5);
    %hxp66(icnt) = h(6,6);
```

```
    zabc11(icnt) = inv(h(1,1));
    %zabc22(icnt) = inv(h(2,2));
    %zabc33(icnt) = inv(h(3,3));
    %zabc44(icnt) = inv(h(4,4));
    %zabc55(icnt) = inv(h(5,5));
    %zabc66(icnt) = inv(h(6,6));
```

```
end
```

```
orient landscape
figure(1)
plot(freq*2*pi,log10(hxp11)), grid
xlabel('Frequency (rad/sec)')
axis([0,2,-2,3])
```

```
figure(gcf+1)
plot(freq*2*pi,log10(zabc11)), grid
xlabel('Frequency(rad/sec)')
axis([0,2,-3,2])
```

## Hmat.m

```
% This program is used for the experimental data obtained using the I-DEAS
% software. It takes the data stored in beamb and creates an FRF matrix as well as
% an inverse FRF matrix for any boundary condition. The user inputs the
% frequency range and the nodes to be used in the a-set.
%
% Written by C.P. DeGregory

clear

load beamb % Loads data file containing complex amplitudes for each frequency data
           % point in the spectrum.
h11 = Hbase(:,1);

% Plot the FRF for the free-free beam (There are 1601 spectral lines: one every 1.25 Hz)

plot(0:1.25:2000,20*log10(abs(h11))),grid;
axis([0,500,-40,50]);

first = input('Enter first frequency: ');
last = input('Enter last frequency: ');

start = (first/1.25) + 1;
finish = (last/1.25) + 1;

% Initialize

icnt = 0;
H = zeros(5*((finish-start)+1),5);
freqp = [first:1.25:last];
Zap = zeros(length(freqp),1);

aset = input('Enter nodes to be used in a-set (vector form): ');

% Build H matrix

for i = (start) : (finish)

    H((5*icnt-4):5*icnt,:) = [Hbase(i,1:5); Hbase(i,6:10); Hbase(i,11:15);...
                             Hbase(i,16:20); Hbase(i,21:25)];

    % Build Haa matrix

    Ha((length(aset)*icnt)-length(aset)+1:(length(aset)*icnt), :) = H ((5*...
```

```

        icnt-1) +aset, aset);
Za((length(aset)*icnt)-length(aset)+1:(length(aset)*icnt), :) = inv(Ha((length...
    (aset)*icnt)-length(aset)+1:(length(aset)*icnt), :));

% 1,1 Element for plotting

Zap(icnt) = Za(((length(aset)*icnt) - length(aset)) +1,1);

end

plot(freqp, 20*log10(abs(Zap))),grid

```

## B. SUPPORT PROGRAMS

### Assemble2Beams.m

```
% This program assembles the global mass and stiffness matrices of the two beams
% defined in "Build2Beams" by combining element k and m matrices solved in the
% function "fbeamkm."
%
% Loop over the two beams:
```

```
for icnt_beams = 1:2;
```

```
    k=zeros(ndof,ndof);
    m=zeros(ndof,ndof);
```

```
    % Loop over the number of elements in the structure:
```

```
    % ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~
```

```
    for elnum=1:num_elements;
```

```
        dof1=2*connect(elnum,1)-1;
        dof2=dof1+1;
        dof3=2*connect(elnum,2)-1;
        dof4=dof3+1;
```

```
        % ... set up beamel function call:
```

```
        l_temp=element_length;    % Using fixed element lengths
        ei_temp=element_EI(elnum,icnt_beams);
        m_temp=element_mass(elnum,icnt_beams);
```

```
        [kbeam,mbeam]=fbeamkm(l_temp,ei_temp,m_temp);
```

```
        k(dof1:dof2,dof1:dof2)=k(dof1:dof2,dof1:dof2)+kbeam(1:2,1:2);
        k(dof1:dof2,dof3:dof4)=k(dof1:dof2,dof3:dof4)+kbeam(1:2,3:4);
        k(dof3:dof4,dof1:dof2)=k(dof3:dof4,dof1:dof2)+kbeam(3:4,1:2);
        k(dof3:dof4,dof3:dof4)=k(dof3:dof4,dof3:dof4)+kbeam(3:4,3:4);
```

```
        m(dof1:dof2,dof1:dof2)=m(dof1:dof2,dof1:dof2)+mbeam(1:2,1:2);
        m(dof1:dof2,dof3:dof4)=m(dof1:dof2,dof3:dof4)+mbeam(1:2,3:4);
        m(dof3:dof4,dof1:dof2)=m(dof3:dof4,dof1:dof2)+mbeam(3:4,1:2);
        m(dof3:dof4,dof3:dof4)=m(dof3:dof4,dof3:dof4)+mbeam(3:4,3:4);
```

```
    % end loop over the number of elements:
```

```

end

% Reassign k and m to new variables and add lumped masses
% ~~~~~

if icnt_beams == 1;
    ka = k;ma = m;
elseif icnt_beams == 2;
    kx = k;mx = m;
end

% End icnt_beams loop:
end

clear dof1 dof2 dof3 dof4 l_temp ei_temp m_temp icnt_beams
clear k m kbeam mbeam elnum

```

## BeamA\_Prompt.m & BeamProperties.m

```
% This program prompts the user for Beam A data. The first inputs required are the
% nominal physical properties. The short program file "BeamProperties" is offered
% to the user. The data listed in "BeamProperties" is the data that was used for all
% beams throughout this thesis.
```

```
disp(' ');disp(' ');
disp(' Enter nominal physical properties for first beam')
disp(' ~~~~~~')
props_file = 'n';
props_file = input(' Run "BeamProperties.m" script to define nominal properties ?...
(y/n) >> ','s');
```

```
if props_file == 'y';
    BeamProperties;
else;
    total_length = input(' Enter the total beam length in inches: ');
    num_elements = input(' Enter the number of elements: ');
    nominal_EI = input(' Enter the nominal EI value (lbf/in^2): ');
    nominal_area = input(' Enter the nominal cross section area (in^2): ');
    nominal_density = input(' Enter the nominal weight density (lbf/in^3): ');
end;
```

```
for icnt_elements = 1:num_elements;
    connect(icnt_elements,1:2) = [icnt_elements,icnt_elements+1];
end
```

```
ndof=length(connect)*2+2; % this is unrestrained beam DOF
```

```
element_length = total_length/num_elements;
element_EI(:,1) = nominal_EI * ones(num_elements,1);
element_area(:,1) = nominal_area * ones(num_elements,1);
element_density(:,1) = nominal_density * ones(num_elements,1);
element_mass(:,1) = (element_density .* element_area * element_length)/386.4;
```

```
% End Data input for 1st beam: Copy data for 2nd beam:
```

```
% ~~~~~~
element_EI(:,2) = element_EI(:,1);
element_area(:,2) = element_area(:,1);
```



```
element_density(:,2) = element_density(:,1);  
element_mass(:,2) = element_mass(:,1);
```

```
% End BeamA_Prompt.m
```

```
% _____  
% _____
```

```
% BeamProperties.m
```

```
% This is the file to load nominal beam data.
```

```
total_length = 60;  
num_elements = 10;  
nominal_EI = 5.5e5;  
nominal_area = .75;  
nominal_density = .283;
```

## BeamX\_Prompt.m

```
% This program prompts the user for Beam X data. Specifically, the user is asked to
% choose whether to input element EI modifications or to have them chosen
% randomly. If chosen randomly, the data is saved in the binary (.mat) file
% "changedata" and is then offered as a choice for the next run of this code. By
% doing this, the option exists to create 2 free-free beams with unknown damage and
% 2 a-set constrained beams with the same (but still unknown) damage.
```

```
disp(' ');disp(' ');
disp(' Modify nominal physical properties for second beam')
disp(' ~~~~~~')

```

```
% Adjust EI values for second beam:
```

```
option = 'u';
option1 = 'n';
change_EI = 'n';
```

```
change_EI = input(' Modify single/range element EI values (y/n)? ','s');
option = input(' User defined, random, or neither (u,r,n)? ','s');
option1 = input(' Read previous modification values (y/n)? ','s');
if option1 == 'y'
    load changedata
    change_EI = 'n';
end
```

```
while (change_EI ~= 'n') & (option == 'u')
```

```
    disp(' Enter element label(s) for EI modification')
    disp(' Use MATLAB vector format> 1 3 5:7 9 ')
    new_lbls = input(' >> ','s');
    new_lbls = eval(['[',new_lbls,']']); % Converts string to vector of labels
```

```
    disp(' Enter EI change for element range')
    EI_change = input(' Enter percentage EI change (+/- %) ');
    element_EI(new_lbls,2) = element_EI(new_lbls,2)+...
    (EI_change/100) * element_EI(new_lbls,2);
```

```
    change_EI = input(' Modify another element EI value (y/n)? ','s');
```

```
end; % end while
```

```

if option == 'r'
    numels = fix(3*rand+1);
    for i=1:numels
        new_lbls = fix(10*rand+1);
        format short
        EI_change = -rand/2;
        element_EI(new_lbls,2) = element_EI(new_lbls,2)+...
            EI_change*element_EI(new_lbls,2);
    end

    save changedata numels new_lbls element_EI
end

% End BeamX_Prompt.m

```

### BeamSens\_Prompt.m

```
%      This program prompts the user for design variable element label defintitions. It is
%      called by the program "Beam_Sensitivity."

max_num_dv = 50;

% ***** EI DV ELEMENT LABEL PROMPTING *****

define_EI_dv = input(' Calculate element EI sensitivities (y/n)? ','s');

if define_EI_dv == 'y';

    all_elements = input(' Define EI DV"s for all elements? (y/n) ','s');
    if all_elements == 'y';
        for icnt_elements = 1:num_elements;
            EI_el_lbl(icnt_elements,1) = icnt_elements;
        end;
        define_EI_dv = 'n';

    else;      % all_elements == no, user selects elements

        for icnt_dv = 1 : max_num_dv;

            sprintf(' Enter element label(s) for EI DV # %3i',icnt_dv)
            disp(' Use MATLAB vector format> 1 3 5:7 9 -- "d" to finish')
            new_lbls = input(' >> ','s');
            if new_lbls ~= 'd';
                new_lbls = eval(['[',new_lbls,']']); % Converts string to
                                                    % vector of labels
                EI_el_lbl(icnt_dv,1:length(new_lbls)) = new_lbls;
            else
                break
            end;

        end; % for icnt_dv
    end;      % if all_elements

    mass_el_lbl = [ ];
end;

% ***** END ELEMENT LABEL PROMPTING *****
% End script.
```

## Boundary\_Conditions.m

```
% This program prompts the user for boundary condition information, creates a  
% vector of DOF (with respect to the unrestrained structure) and then extracts the  
% rows and columns of the complementary DOF. The program defines vector  
% "free_dof_set" containing the list of unrestrained DOF. The boundary conditions  
% for the given system are applied in this script.
```

```
if exist('free_dof_set')==0; % Build free_dof_set vector
```

```
    disp(' Select a boundary condition set:')  
    disp(' (1) Clamped-free')  
    disp(' (2) Clamped-Clamped')  
    disp(' (3) Pinned-Pinned')  
    disp(' (4) User-Defined')  
    disp(' (5) Free-Free')
```

```
    BC_Choice = input(' >> Enter choice: ');
```

```
    if BC_Choice == 1; % Clamped-free
```

```
        free_dof_set = [3:ndof];  
        restraint_switch = 'y';
```

```
    elseif BC_Choice == 2; % Clamped-Clamped
```

```
        free_dof_set = [3:ndof-2];  
        restraint_switch = 'y';
```

```
    elseif BC_Choice == 3; % Pinned-Pinned
```

```
        free_dof_set = [2:ndof-2 ndof];  
        restraint_switch = 'y';
```

```
    elseif BC_Choice == 4; % User-Defined
```

```
        icnt_dof = 0;  
        add_dof = 'y';  
        while add_dof == 'y';  
            bc_node = input(' Node number for restraint ? "0" to end: ');  
            if bc_node == 0;  
                break  
            end;  
            bc_coord = input(' Translation or Rotation ? (t/r) ','s');
```

```

        icnt_dof = icnt_dof + 1;
        if bc_coord == 't';
            bc_DOF(icnt_dof) = 2 * bc_node - 1;
        elseif mass_coord == 'r';
            bc_DOF(icnt_dof) = 2 * bc_node;
        end; % End if-elseif block

    end; % End while add_dof

    bc_boolean = ones(ndof,1); % [1 1 1 ... icnt_dof]
    bc_boolean(bc_DOF) = zeros(length(bc_DOF),1); % Put zeros in restrained dof
    all_dofs = [1:ndof]; % List of all dof
    free_dof_set = all_dofs.*bc_boolean'; % Extract free dof
    restraint_switch = 'y';

elseif BC_Choice == 5; % Free-free beam

    free_dof_set = [1:ndof];
    restraint_switch = 'n';

end; % End if-elseif choice block _____

end; % End exist block

free_dof_set(free_dof_set==0)=[ ];

ka = ka(free_dof_set,free_dof_set);
ma = ma(free_dof_set,free_dof_set);
kx = kx(free_dof_set,free_dof_set);
mx = mx(free_dof_set,free_dof_set);

% ***** END BOUNDARY_CONDITIONS.M *****

```

## C. FUNCTIONS CALLED BY MAIN & SUPPORT PROGRAMS

**function[kbeam,mbeam] = fbeamkm(l\_temp,ei\_temp,m\_temp)**

% This function computes the elemental stiffness and mass matrices defined using  
% Galerkin's Method and Hermitian shape functions. [Ref 7]

```
krow1 = [12 6*l_temp -12 6*l_temp];  
krow2 = [6*l_temp 4*l_temp^2 -6*l_temp 2*l_temp^2];  
krow3 = [-12 -6*l_temp 12 -6*l_temp];  
krow4 = [6*l_temp 2*l_temp^2 -6*l_temp 4*l_temp^2];  
kbeam = (ei_temp/l_temp^3)*[krow1;krow2;krow3;krow4];
```

```
mrow1 = [156 22*l_temp 54 -13*l_temp];  
mrow2 = [22*l_temp 4*l_temp^2 13*l_temp -3*l_temp^2];  
mrow3 = [54 13*l_temp 156 -22*l_temp];  
mrow4 = [-13*l_temp -3*l_temp^2 -22*l_temp 4*l_temp^2];  
mbeam = (m_temp/420)*[mrow1;mrow2;mrow3;mrow4];
```

**function [phi\_normal,orth]=massnormal(phi,mass)**

% This function mass normalizes a modal matrix.

```
a=size(phi);  
nummodes=a(1,2);  
phi_normal=zeros(size(phi));  
%  
for ii=1:nummodes;  
    modalmass(ii)=phi(:,ii)'*mass*phi(:,ii);  
    if modalmass(ii)~=0,  
        phi_normal(:,ii)=(1/sqrt(modalmass(ii)))*phi(:,ii);  
    else  
        phi_normal(:,ii)=phi(:,ii);  
    end  
end  
%  
% do the orthogonality calculation:  
%  
orth=phi_normal'*mass*phi_normal*100;
```

**function [omega,phi]=fmodes(k,m)**

% This function returns a vector containing eigenvalues  $(\text{rad/sec})^2$  and a matrix  
 % containing the mass normalized mode shapes. The mode information is sorted by  
 % frequency in ascending order.

[v,d]=eig(m\k);

[omega,index]=sort(diag(d));

[phi,orth]=fmassnormal(v(:,index),m);

format long

disp('Freqs in Hz.:')

disp('~~~~~')

disp(sqrt(omega(1:min(length(omega),12)))/2/pi)

disp('-----')

**function [oset] = fOset\_from\_Aset(ndof,aset);**

% This function determines the complementary o-set from a set [1:1:ndof] and the  
 % subset a-set = [x x x ...].

%

% ndof: Total number of DOF. Set is labeled "nset".

% aset: Retained DOF (proper subset of [1:1:ndof])

nset = [1:ndof];

for icnt = 1 : length(aset);

    indices(icnt) = find(nset == aset(icnt));

end

bool = ones(size(nset));

bool(indices) = zeros(size(indices));

oset = nset(find(bool>0));



**function[k,m]=fspringmass(conn,mspec,fix)**

% This function assembles the stiffness [k] and mass [m] matrices for an assemblage of springs. The connectivity is established in the array 'conn', the boundary conditions (fixity) are specified in the array 'fix'. The mass is specified in the vector mspec.

% For example, to build the following system:

```
%
%      m      2m      m
%      |---|---|---|---|---|
%      k      k      2k
```

% the connectivity array is as follows:

```
%      [1 2
%       2 3
%       3 4
%       3 4]
```

% The connectivity array is therefore of dimension:  
(number of springs)\*(2).

% The fixity array for the above system is

```
%      fix = [0 1 1 1]
```

% The fixity array is therefore of dimension:  
(1)\*(number of nodes).

% The mass distribution is given by mspec:

```
%      mspec = [1 2 1]
```

% The mspec array is of dimension:  
(1)\*number of dynamic (free) DOF

```
kspring=[1 -1;-1 1];
ksize=length(fix);
k=zeros(ksize,ksize);
m=zeros(length(mspec));
```

% assemble stiffness matrix:

```
for spring = 1:size(conn,1);
    index1=conn(spring,1);
    index2=conn(spring,2);
```

```

        k(index1,index1)=k(index1,index1)+kspring(1,1);
        k(index1,index2)=k(index1,index2)+kspring(1,2);
        k(index2,index1)=k(index2,index1)+kspring(2,1);
        k(index2,index2)=k(index2,index2)+kspring(2,2);
    end;

    % zero rows and columns corresponding to boundary conditions

    indices_to_keep = find(fix);

    if length(indices_to_keep) ~= 0;
        k = k(indices_to_keep,indices_to_keep);
    end

    %assemble mass matrix:

    m=zeros(size(k));
    for i=1:length(m);
        m(i,i)=mspec(i);
    end

```

## LIST OF REFERENCES

1. Gordis, J.H. "Artificial Boundary Conditions for Model Updating and Damage Detection." To appear, *Journal of Mechanical Systems and Signal Processing*.
2. Gordis, J.H. "Omitted Coordinate Systems and Artificial Constraints in Spatially Incomplete Identification." *Modal Analysis*, v.11 n1 July 1996, pp. 83-95.
3. Gordis, J.H. "An Analysis of the Improved Reduced System (IRS) Model Reduction Procedure." *Modal Analysis*, v. 9 n4 October 1994, pp. 269-285.
4. Li, S., Shelley, S., Brown, D. "Perturbed Boundary Condition Testing." *Proceedings of the 13<sup>th</sup> International Modal Analysis Conference*, v. 1, 1995, pp. 902-907.
5. Hemez, F.M. "Closing the Gap Between Modal Parameter Based and Frequency Response Function Based Updating Methods." *Proceedings of the 13<sup>th</sup> International Modal Analysis Conference*, v. 1, 1995, pp. 171-177.
6. Inman, D.J. *VIBRATION With Control, Measurement, and Stability*, pp. 129-131, Prentice-Hall, Inc., 1989.
7. Kwon, Y.W., Bang, H., *The Finite Element Method Using MATLAB*, pp. 235-242, CRC Press Inc., 1997.



## INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center .....	2
8725 John J. Kingman Road, Ste 0944	
Ft. Belvoir, VA 22060-6218	
2. Dudley Knox Library .....	2
Naval Postgraduate School	
411 Dyer Road	
Monterey, CA 93943-5101	
3. Professor J.H. Gordis, Code ME/Go.....	4
Department of Mechanical Engineering	
Naval Postgraduate School	
Monterey, CA 93943	
4. Department Chairman, Code ME .....	1
Department of Mechanical Engineering	
Naval Postgraduate School	
Monterey, CA 93943	
5. Naval Engineering Curricular Office (Code 34).....	1
Naval Postgraduate School	
Monterey, CA 93943	
6. LT Christopher P. DeGregory.....	1
173 Bridle Path Circle	
Ludlow, MA 01056	